Learning Group Composition and Re-composition in Large-scale Online Learning Contexts

Dissertation

zur Erlangung des akademischen Grades
doctor rerum naturalium
(Dr. rer. nat.)
im Fach Informatik

eingereicht an der
Mathematisch-Naturwissenschaftlichen Fakultät
der Humboldt-Universität zu Berlin

von

Zhilin Zheng, M. Eng.

Präsidentin der Humboldt-Universität zu Berlin:

Prof. Dr.-Ing. habil. Dr. Sabine Kunst

Dekan der Mathematisch-Naturwissenschaftlichen Fakultät:

Prof. Dr. Elmar Kulke

Gutachter:

1. Prof. Dr. Niels Pinkwart
2. Prof. Dr. Johannes Konert
3. Prof. Dr. Christoph Meinel

Tag der mündlichen Prüfung: 28.08.2017

ABSTRACT

Small learning group composition addresses the problem of seeking such matching among a population of students that it could bring each group optimal benefits. Recently, many studies have been conducted to address this small group composition problem. Nevertheless, the focus of such a body of research has rarely been cast to large-scale contexts. Due to the recent come of MOOCs, the topic of group composition needs to be accordingly extended with new investigations in such large learning contexts. Different from classroom settings, the reported high drop-out rate of MOOCs could result in group's incompletion in size and thus might compel many students to compose new groups. Thus, in addition to group composition, group re-composition as a new topic needs to be studied in current large-scale learning contexts as well.

In this thesis, the research is structured in two stages. The first stage is group composition. In this part, I proposed a discrete-PSO algorithm to compose small learning groups and compared the existing group composition algorithms from the perspectives of time cost and grouping quality. To implement group composition in MOOCs, a group composition experiment was conducted in a MOOC. The main results indicate that group composition can reduce drop-out rate, yet has a very weak association with students' learning performance. The second stage is to cope with group re-composition. This thesis suggests a data-driven approach that makes full use of group interaction data and accounts for group dynamics. Through evaluation in a simulation experiment, it shows its advantages of bringing us more cohesive learning groups and reducing the drop-out rate compared to a random condition. Apart from these, a group learning tool that fulfills the goals of the proposed group re-composition approach has been developed and is made ready for practice.

ZUSAMMENFASSUNG

Die Erforschung der Zusammenstellung kleiner Lerngruppen beschäftigt sich mit dem Problem, eine passende Gruppenzusammensetzung in einer Population von Lernern zu finden, die jeder Gruppe optimalen Nutzen bringen könnte. In letzter Zeit sind viele Studien zu diesem Problem der Kleingruppenzusammenstellung durchgeführt worden. Allerdings waren diese Forschungen nur selten auf den Kontext großer Lerner-Populationen ausgerichtet. Angesichts des zunehmenden Aufkommens von MOOCs muss jedoch das Problem der Gruppenzusammenstellung entsprechend erweitert betrachtet werden, und zwar mit neuen Forschungen, die den Kontext derartig großer Lerner-Populationen berücksichtigen. Anders als in Klassenzimmer-Settings könnte die beobachtete hohe Abbruchquote in MOOCs in einer Unterbesetzung der Gruppengröße resultieren und könnte somit viele Lerner dazu bringen, neue Gruppen zu bilden. Zusätzlich zur Gruppenzusammenstellung muss daher die Gruppenneuzusammenstellung als neues Thema in aktuellen Kontexten großer Lerner-Populationen ebenfalls erforscht werden.

Die Untersuchungen der vorliegenden Arbeit gliedern sich in zwei Teile. Der erste Teil beschäftigt sich mit Gruppenzusammenstellung. In diesem Teil stelle ich einen diskreten-PSO Algorithmus zur Zusammenstellung kleiner Lerngruppen vor und vergleiche bislang bestehende Gruppenzusammenstellungs-Algorithmen unter den Gesichtspunkten Zeitaufwand und Gruppierungsqualität. Um Gruppenzusammenstellung in MOOCs anzuwenden wurde ein Gruppenzusammenstellungsexperiment in einem MOOC durchgeführt. Die Hauptergebnisse deuten darauf hin, dass die Gruppenzusammenstellung die Abbruchsquote reduzieren kann, jedoch lediglich einen sehr schwachen Bezug zur Lernperformanz der Lerner aufweist. Der zweite Teil beschäftigt sich mit Gruppenneuzusammenstellung. Die vorliegende Arbeit stellt eine datengesteuerte Herangehensweise vor, die umfassenden Gebrauch von Gruppeninteraktionsdaten macht sowie Gruppendynamik mit einbezieht. Mittels einer in einem Simulationsexperiment durchgeführten Evaluation zeigen sich die Vorteile dieses Verfahrens: Der Lerngruppenzusammenhalt wird verbessert und die Abbruchsquote im Vergleich zu einer Zufallsverteilung reduziert. Darüberhinaus wurde hier ein Gruppen-Lern-Werkzeug entwickelt und für die Praxis vorbereitet, das die Anforderungen des geforderten Ansatzes der Gruppenneuzusammenstellung erfüllt.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my supervisor, Prof. Dr. Niels Pinkwart, for his advice and guidance on my thesis work throughout these years. From Clausthal-Zellerfeld to Berlin, we have made countless weekly meetings, at which many of our ideas have been exchanged. Without his trust and support, I would never have stepped into the Educational Data Mining research field and thus would have never made any contributions to this fantastic research area.

The living expenses during my PhD study were covered by a Chinese scholarship. I am grateful to the scholarship provider, the China Scholarship Council (CSC), for giving me an opportunity to study overseas.

Many thanks go to my thesis reviewers, Prof. Dr. Christoph Meinel and Prof. Dr. Johannes Konert, for their effort to review my thesis and their valuable feedback.

I am grateful to the students and co-authors who related to this thesis, Jan Bundrock, Martin Stapel and Tim Vogelsang. Their contributions absolutely brighten this thesis. I should also thank Nguyen-Thinh Le, Andreas Lingnau, Hanna Rochlitz and Evy Keeney-Ritchie for language checking of this thesis. I would also like to thank all of my colleagues from the CSES research group for creating a very friendly working environment.

My wife and children, Geyu, Jayden and Elisa, gave me much love and support during the course of my PhD study, and I thank them. Although my mother, father, mother-in-law and father-in-law stay in distant China, they also deserve my gratitude for their constant support and understanding.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

1 INTRODUCTION

三人行，必有我師焉。擇其善者而從之，其不善者而改之。

Confucius

Confucius said: "*when three men walk together, there is always something I can learn. Choose to follow what is good in them and correct what is not good*"[1]. Learning is not merely doing something in isolation, but also it comes from social activities. As such, many teachers, in the modern educational system, advocate and practice peer education in schools and colleges. For example, they sometimes assign students into small learning groups in which students are expected to exchange their knowledge and learn skills from one another. Nevertheless, creating such small groups that can bring great learning is not easy.

1.1 Motivation

Dating back to the year of 2012, I was setting out my doctoral study from scratch – no topic and many headaches. In that very year, the New York Times proclaimed it the "year of the MOOC"[2] since MOOCs (Massive Open Online Courses) have started a trend and began a new era of online education. It is not clear how big of an audience

[1] http://www.confucius.org/lunyu/ed0721.htm

[2] http://www.nytimes.com/2012/11/04/education/edlife/massive-open-online-courses-are-multiplying-at-a-rapid-pace.html

this report has attracted, but most importantly, it came into my view and led my thinking to that growing field – MOOCs.

Differing from the traditional classroom and e-learning systems, the relatively big scale is one of the MOOCs' selling points. Those, either non-profit or for-profit, MOOC platforms (e.g. Coursera[3], edX[4] and Udacity[5]) scale up the online courses and bring them to massive users across the world via Internet. One course could unprecedentedly enroll thousands of online students. The courses recently published on those platforms cover a variety of disciplines (e.g. Maths, History, Business, Arts and Humanities and Computer Science). Besides, those world-class courses are no longer only brought to a very small number of institutions but also to every cyber citizen and can thus democratize education to some extent.

MOOCs, to some extent, can be considered as one of the best practices of the *connectivism* theory. The *connectivism* theory emphasizes acquisition of organizational knowledge via connecting information entities (Downes, 2008; Siemens, 2004, 2006a, 2006b, 2014). As Siemens states in (Siemens, 2014, p. 5), "*Learning (defined as actionable knowledge) can reside outside of ourselves (within an organization or a database), is focused on connecting specialized information sets, and the connections that enable us to learn more are more important than our current state of knowing*". Many *connectivism* believers may envision that MOOCs bring our education to such a large community that social connections could take place very often. However, the recent reports seem to broadcast a different picture. The completion rate is reported to be less than 13 percent in most MOOCs (Jordan, 2014) and only 5-10 percent of all learners actively participate in online course forums (Rosé & Siemens, 2014). Obviously the social connection is currently far away from the full-load running mode.

There could be a variety of reasons for dropout (e.g. no real intention to complete, lack of time and so forth) (Onah, Sinclair, & Boyatt, 2014). Participants often lack a persistently workable time plan to follow their registered MOOC courses. They could be disturbed by some other abrupt matters in their daily life. They could also simply

[3] https://www.coursera.org/

[4] https://www.edx.org/

[5] https://www.udacity.com/

find their high motivation fade faster as the course goes further, so much so that they lose interest at all.

In addition, we should not ignore the effect of the weak social ties among participants. In accord with an empirical study, they found that stronger social connections are predictive of the lower drop-out (Yang, Wen, & Rose, 2014). This could imply that promoting social connections would probably reduce the drop-out rate. Before thinking how to promote social connections in MOOCs, let us address another fundamental question: what hurts the emergence of massive social connections? Massive students could have made a tremendous number of social connections. Why is the fact not in support of such in the end? The answer might be massiveness itself. In effect, massiveness is a two-edge sword. It, on the one hand, certainly brings about the potential to share massive course-related learning resources and diverse ideas from a large number of students. But this is based on an assumption that online students are ready to be social. If they have not fully prepared yet, facing such massive peer students would, on the other hand, bring them much more stress rather than opportunity. As Sharan quoted in (Sharan, 1994, p. 220), many students might not be comfortable when facing such a big online audience in such a big community.

> *Moving too quickly towards expecting pupils to talk openly in a large group is rather like throwing a whole class into the deep end of a swimming pool. As well as the few non-swimmers who might drown (never to be heard from again?!), there will be others who are unhappy out of that depth. Whilst a few strong, natural swimmers splash about happily, the nervous ones are so anxious about drowning, or so conscious and embarrassed about their lack of fineness as swimmers, that they lose all confidence and cling silently on to the side of the pool. They need to be brought gradually up to the deep end (Howe, 1988, p.34).*

Small learning groups (of relatively smaller size) might be able to relieve the aforementioned stress that a big online audience could cause. Meanwhile, they can foster social connections as well. As Stahl pointed out in his publication (Stahl, 2015, p. 19), when describing learning, we can put it into three different levels: individual level, group level and community level. The recent MOOCs meet students' individual endeavor need and also try to deliver the community benefits via a public course forum, which obviously covers both individual level and community level. However the group level had not been explored. This, in the end, inspired me to create small learning groups in MOOCs in the hope of filling the gap and bringing group learning experience to MOOC students.

1.2 Research questions

Group learning is such a practical pedagogy that it is often employed to help learners with building up and correcting their own knowledge structure through discussion in small groups as the Constructive theory practitioners pointed out (Anderson & Dron, 2010). Recent research results from the CSCL (Computer Supported Collaborative Learning) community suggest that adding a specifically designed groupware platform with learning oriented features can encourage more social interaction, critical thinking and thus promote deep learning (Newman, Webb, & Cochrane, 1995). Applying this strategy to the current MOOCs, some may argue that it would largely shrink the large community into very small cohorts and thus limits the potentially large social networking. However, it depends on how to use this strategy. If it is applied as a complement to the current course forum in MOOCs, participants can still connect to their massive fellows as usually as they do. Additionally, if the small group learning allows learners to change their groups, they would be able to build up networking with as many group members as they can. As such, the massive social contact could still keep intact.

The research on small-group learning is not new. Dating back to the 1950s, researchers began to do within-class grouping, between-class grouping and even cross-grade grouping. By then, they often composed students into groups according to their achievement, attainments and aptitudes. In general, the students were assigned into homogenous or heterogeneous groups. Up to now, the grouping features have been growingly enriched. For instance, learning style, demographic characteristics, and behavioral attributes have been explored by many researchers already.

Nevertheless, MOOCs, as a trending means to deliver education, bring many more new challenges than the classroom teaching or e-learning. Blending small learning groups into such a new learning environment therefore needs to be done very carefully rather than by applying a simple copy mode. For instance, does the unprecedented large number of students challenge the creation of small learning groups? Thousands of students or even tens of thousands of students is the norm in MOOCs. Assigning this large number of students would probably costs much more in computation than dozens of students in classrooms or hundreds of students in e-learning systems, for instance. What new problems could arise in small learning groups when the learning environment is completely open? '*Open*' means that everyone can join the course as long as it is still ongoing and everyone can arbitrarily leave, which in classrooms rarely occurs. Along this line of thought, there should be many more interesting questions to ask. In this thesis, the focus however centers around the creation of small learning groups in MOOCs. The main questions addressed in this thesis can be summarized as follows:

- **RQ1:** What methods are employed to compose small learning groups, and which out of those group composition methods could be suitable for large-scale learning settings and what about their efficiency?
- **RQ2:** How can one apply a group composition method to a MOOC course and what is the impact of group composition on dropout and learning performance?
- **RQ3:** When creating small learning groups in the large scale learning settings (e.g. MOOCs), what new problems could arise and is there an approach to mitigate those problems?
- **RQ4:** If there is an approach, how can one put it into practice and what benefits could this approach bring?

1.3 Research approach

To address the research questions in Section 1.2, the research approaches applied in this work include reviewing literature, proposing grouping algorithms, developing software and conducting computer simulation experiments.

RQ1: relevant literature on the group composition methods that have been applied in recent decades was reviewed. To fill a research gap, a discrete-PSO algorithm was proposed to compose small learning groups. In addition, four typical group composition algorithms were tested with a MOOC dataset. Based on the experimental results, their advantages together with pitfalls are discussed.

RQ2: an experiment was conducted in a real MOOC course. In this experiment, a group composition method was applied to compose small learning groups in MOOCs. The method is easy to be replicated in other MOOC courses. Compared to a random grouping condition and a condition of no grouping, the experimental results indicate that using this method can reduce the dropout rate, yet it has little impact on the learning performance.

RQ3: based on relevant literature and empirical evidence collected from a MOOC experiment, I saw the problems that could arise if creating small learning groups in MOOCs. To mitigate those problems, a dynamic group re-composition approach was proposed accordingly.

RQ4: putting the proposed group re-composition approach into practice requires a small-group learning tool. To the best of my knowledge, such a tool has not been developed by the mainstream MOOC platforms, nor by my industry partner, iversity[6]. Due to this fact, there was no better choice than developing a group tool that enables the experimental requirements of the group re-composition approach. This group tool offers such main components as collaborative writing, peer grading, group composition and group re-composition. With such a group tool, the implementation of the proposed approach can be technically addressed. Moreover, to examine benefits that the dynamic group re-composition can bring, a simulation experiment was conducted. Through observing the drop-out and group cohesion, the experimental results indicate its positive impact.

1.4 Organization of the thesis

This thesis is structured into six chapters as shown in Figure 1.1. Each chapter is briefly described as follows:

Chapter 2 presents related work that has recently been done. It starts with methods to solve the classical group composition problem. The focus is its long historical development and broad application scenarios. It then continues with group composition work that has recently been done in MOOCs. Lastly, it reviews some work on group re-composition in recent years.

Chapter 3 starts with mathematical modelling of the group composition problem. It then proposes a discrete particle swarm optimization algorithm followed by testing its performance in computer simulation. To examine the proposed algorithm's advantages over several of its counterparts (i.e. ant colony optimization, genetic algorithm and a k-means variant algorithm), the thesis simulates and compares their performance in a

[6] https://iversity.org/

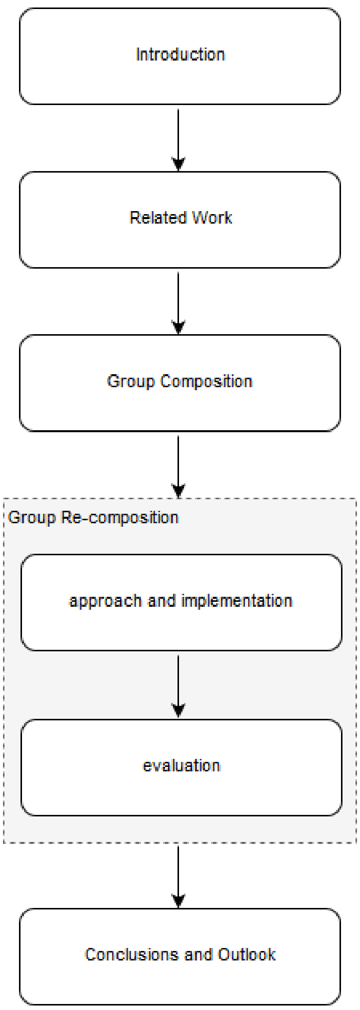
same testing bed. It ends with an experiment conducted in a MOOC course. The research questions RQ1 and RQ2 are addressed until this chapter.

Chapter 4 begins to discuss the problems that would arise if the existing group composition methods are employed in MOOCs. It then elaborates a newly proposed group re-composition approach that attempts to mitigate those problems. It ends with the design and development of a group tool aiming at putting the group re-composition approach into practice. Until this chapter, the research question RQ3 is addressed.

Chapter 5 describes a computer simulation conducted to validate the group re-composition approach and examine its impact on both the drop-out rate and group cohesion. The answers to the research question RQ4 can be found in this chapter.

Chapter 6 concludes the thesis, discusses the findings and lessons learned, reflects some improvements that need to be done in the future work, and envisions future trends to pursue the current work.

Figure 1.1: Thesis Outline



2 RELATED WORK

This chapter gives a synopsis of existing research work related to the topic of group composition and re-composition. With regard to the topic of group composition, plenty of research has been done. A common goal of those works is simple: to compose students into small learning groups. However, they vary a lot from the perspectives of grouping factors (e.g. personalities and team roles), grouping criteria (e.g. ability heterogeneous groups and background knowledge homogeneous groups) and grouping methods (e.g. random methods and algorithmic methods). This chapter seeks to outline the categories that those works could best situate and attempts to bring a better understanding to them. In recent years, some researchers have begun to experiment on group composition in MOOCs. However, the topic of group re-composition has rarely been explored.

2.1 Factors in effective groups

Seeking the factors contributing to effective groups is of importance. Much research has been conducted to address this. The following will unveil some evidence from the recent scientific reports. The narrative accounts for two aspects: 1) what factors have been suggested by the empirical studies? 2) what factors have been used by the grouping practitioners?

2.1.1 What is empirically suggested?

Recent empirical studies, particularly from the fields of psychology and business, have increasingly suggested to us the factors of effective groups. Koppenhaver et al. found the importance of team motivation and stability in a context of instructor-assigned teams (Koppenhaver & Shrader, 2003). Whittingham's findings suggest the significant impact of students' personality on learning performance of MBA students (Whittingham, 2006). Applying team roles theory has also been reported to positively impact learning performance (Senior, 1997; Yannibelli & Amandi, 2012b). A review work on group interaction and learning given by Webb concludes that such interactive behaviors as giving help and receiving help make positive contributions to group achievement whereas off-task behaviors exert a negative influence (Webb, 1982). Students' ability level has also been studied in small groups. Beane et al. drew a conclusion that low-performing students benefit from homogeneous grouping (Beane & Lemke, 1971). An investigation of Bradley et al. unveils the important role of the composition of personality in successful group performance (Bradley & Hebert, 1997). Stevens also found the importance of Belbin roles in software development teams (Stevens Jr, 1998). Through reviewing a wealth of research on cooperative learning, Slavin concluded that group reward and individual accountability are crucial to the effectiveness of cooperative learning (R. E. Slavin, 1983).

Table 2.1 Factors suggested by empirical studies

Category	**Factor**
Individual-level characteristics	Ability level (Beane & Lemke, 1971)
	Personality characteristics (Bradley & Hebert, 1997; Whittingham, 2006)
	Team roles (Senior, 1997; Stevens Jr, 1998)
	Individual accountability (R. E. Slavin, 1983)
Group-level characteristics	Team motivation (Koppenhaver & Shrader, 2003)
	Team stability (Koppenhaver & Shrader, 2003)
	Interactive behaviors (Webb, 1982)
Instructional behaviors	Group reward (R. E. Slavin, 1983)

Table 2.2 Factors applied in recent grouping works

Category	Factor
Individual-level characteristics	Ability level (Bergey & King, 2014; Feng, Shibin, Cheng, & Qinghua, 2008; C.-C. Hsu, Chen, Huang, Huang, & Huang, 2014; Ounnas, Davis, & Millard, 2009)
	Personality characteristics (Balmaceda, Schiaffino, & Pace, 2014; Bergey & King, 2014; Feng et al., 2008; Srba & Bielikova, 2014)
	Team roles (Balmaceda et al., 2014; Yannibelli & Amandi, 2012a)
	Communicative skills (Moreno, Ovalle, & Vicari, 2012)
	Demographic data (Bergey & King, 2014; Ounnas et al., 2009)
	Learning style (Paredes, Ortigosa, & Rodriguez, 2010)
	Leadership (Moreno et al., 2012)
Group-level characteristics	Group interaction (Srba & Bielikova, 2014)

2.1.2 What is practically applied?

When moving our focus onto what factors have factually been taken into account by the practitioners to compose learning groups, ability-oriented characteristics (e.g. learning performance, previous marks and background knowledge) have been widely taken into consideration (Bergey & King, 2014; Feng et al., 2008; C.-C. Hsu et al., 2014; Ounnas et al., 2009). Team roles have also been used to compose learning groups (Balmaceda et al., 2014; Yannibelli & Amandi, 2012a). Still, many favor personality characteristics (Balmaceda et al., 2014; Bergey & King, 2014; Feng et al., 2008; Srba & Bielikova, 2014). Besides, communicative skills are also very interesting to some (Moreno et al., 2012; Srba & Bielikova, 2014). Note that communicative skills can be retrieved by answering questionnaires before grouping, which turns out to reflect students' individual characteristics (Moreno et al., 2012). They can also be analyzed based on group interaction, which on the other hand reflects group process. Those communicative skills retrieved during group process could probably change as the group develops (Srba & Bielikova, 2014). Apart from these, some other characteristics can be found in recent studies, such as demographic data (e.g. gender and ethnic background) (Bergey & King, 2014; Ounnas et al., 2009), learning style (Paredes et al., 2010) and leadership (Moreno et al., 2012).

By comparing the factors retrieved from empirical studies (cf. Table 2.1) with ones that were practically applied in the grouping operations (cf. Table 2.2), we may have a straightforward impression that individual-level data are fairly interesting to the practitioners, particularly ability level, personality characteristics and team roles. Such attributes are normally obtained via either questionnaires or pre-tests before grouping. For example, in Balmaceda's work (Balmaceda et al., 2014), a well-known Myers-Briggs questionnaire was sent out to collect the students' personality characteristics data. A common feature of these attributes is their static property in group process, that is, presumably, they do not change over time. Yet, such static data could not factually stay static in some cases. For example, students' ability level could rise and fall if we extend the lifecycle of group work to a relatively long period. This, of course, from another side, reveals a shortcoming of using such data. Group-level characteristics, on the other hand, do not have such a problem. Rather, they better reflect group dynamics in the sense that they are retrieved during group process and do not assume any static property of any attribute. For instance, in Srba's work (Srba & Bielikova, 2014), they inferred students' communicative skills by quantitatively analyzing their actions in group interaction(e.g. write comments and give explanations). Gathering such data however relies on the recent automatic data analysis technologies rather than the conventional questionnaires, which, to some extent, hinders its widespread use. Due to the recent advancement of such technologies as Social Network Analysis (SNA) and Natural Language Processes (NLP), dynamic data can be yielded and has thus been increasingly interesting to many researchers and professionals in recent years. In the future, this trend could probably shift the research attention from reliance on the static data to the growing use of group dynamic data.

2.2 Grouping criteria

Grouping criteria are of great importance in group formation too. Along the line of group formation research, a variety of grouping criteria have been used in group creation. Moreno et al. composed heterogeneous ability groups (Graf & Bekele, 2006; Moreno et al., 2012). Heterogeneous learning styles and team roles were also regarded as the group criteria (Balmaceda et al., 2014; Yannibelli & Amandi, 2012a). It is interesting to see that some created personality heterogeneous groups (Balmaceda et al., 2014; Bergey & King, 2014) whereas others, in other studies (Feng et al., 2008; Srba & Bielikova, 2014), favored personality homogeneous groups. There has been much research revealing that personality traits play an important role in predicting group performance (Barrick, Stewart, Neubert, & Mount, 1998; Morgeson, Reider, & Campion, 2005; Whittingham, 2006), but those works do not suggest whether homogeneous groups are superior to heterogeneous groups or not. This leaves grouping practitioners much room to adapt it into their own application scenarios and thus bring some differences from one scenario to another. That is the reason why we saw personality homogeneous groups and heterogeneous groups made in different scenarios. In other words, such grouping criteria are very sensitive to the specific application scenarios. This thesis thus names them application-oriented criteria. Those application-oriented criteria are understandably commonplace in the real world. For example, in international classes, aside from considering knowledge level, students from different cultures may be composed into one group for the purpose of faster cultural integration. But single-culture groups, on the other hand, may drive more efficient learning since group members do not need to spend much time in dealing with such issues as cultural recognition and cultural conflicts. Nevertheless, few yet some grouping criteria are drawn from the existing theories, such as learning styles heterogeneous groups (Felder & Silverman, 1988; Paredes et al., 2010) and Belbin roles heterogeneous groups (Belbin, 1981, 2010; Yannibelli & Amandi, 2012a, 2012b). These are thereby named theory-backed criteria in this thesis.

From the perspective of grouping results, grouping criteria can be categorized into heterogeneous, homogeneous and threshold-based grouping. As we already saw, Belbin roles and learning styles were made heterogeneously in groups (Paredes et al., 2010; Yannibelli & Amandi, 2012a), while background knowledge was homogeneously set up (C.-C. Hsu et al., 2014). Please note that some cases may need a

Table 2.3 Grouping criteria

	Attributes	Theory-backed	Application-oriented	Homogeneous	Heterogeneous	Threshold-based	Single-criterion	Multi-criteria
Balmaceda et al.(Balmaceda et al., 2014)	- team roles - Personality		√			√		√
Srba et al. (Srba & Bielikova, 2014)	- collaborative skills - Personality		√	√				√
Team Machine (Bergey & King, 2014)	- personality traits - demographic variables - years of working experience		√		√			√
Yannibelli et al. (Yannibelli & Amandi, 2012a)	- Team roles	√			√		√	
Moreno et al. (Moreno et al., 2012)	- knowledge levels - communicative skills - leadership skills		√		√			√
Graf et al. (Graf & Bekele, 2006)	- learning performance		√		√		√	
Paredes et al. (Paredes et al., 2010)	- learning style	√			√		√	
Hsu et al. (C.-C. Hsu et al., 2014)	- background knowledge		√	√			√	
Tian et al. (Feng et al., 2008)	- Personalities - Learning performance		√	√				√

mixture of homogeneous and heterogeneous criteria (i.e. with some attributes homogeneous and others heterogeneous) (Gogoulou, Gouli, Boas, Liakou, & Grigoriadou, 2007). Additionally, there is a third category along this dimension, namely, threshold-based criteria. For example, in (Balmaceda et al., 2014), the number of group members who play the same role was limited to up to half of the group size. In the same sense that many commercial teams might not need two group leaders.

Increasingly, many groups have been made to meet multiple criteria rather than a single criterion. This, on the one hand, relies on the multi-modal data retrieval. Except for traditional demographic data and grades available in schools, personality traits, for instance, are no longer difficult to be assessed as long as students fill out psychological questionnaires (e.g. NEO Five-Factor Inventory (NEO-FFI-3)[7] and the Myers&Briggs foundation[8]). On the other hand, it benefits from the emerging computer algorithms that are computationally possible to solve such multi-criteria problems.

2.3 Grouping methods

Over decades, many methods have been proposed to compose learning groups. Random and self-selection methods were firstly adopted in classrooms. As more and more factors and criteria to effective groups have recently been unveiled, many have begun to leverage that knowledge when creating learning groups. Yet, applying those factors and criteria to group creation is computationally expensive. Human eyes and pencils may not be able to handle this task. Some support from computers is rather necessary. As such, up to today, many computer algorithms have been proposed in order to solve the group composition problem efficiently. The following will elaborate these existing methods.

2.3.1 Random methods and self-selection methods

In traditional classrooms, teachers often use random methods and self-selection methods to assign their students into small groups. According to Decker et al.'s study on large-scale business simulations, 52% out of 40 interviewed instructors used self-selection methods to group students and 10% used random assignments (Decker, 1995). Technically, random groups can be composed as Bacon et al. suggested: "*Usually, players are asked to count off by the number of teams and then form a team with other players with the same number.*" (Bacon, Stewart, & Anderson, 2001, p. 7). Random groups seem to provide all participants with fairness. At least, participants

---

[7] http://www.sigmaassessmentsystems.com/assessments/neo-five-factor-inventory-3/

[8] http://www.myersbriggs.org/

have the same likelihood to be assigned to work with others. However scholars (Bacon et al., 2001, p. 8) pointed out that "*Random assignment to teams may also produce too much or too little diversity.*". It can literally be interpreted that many groups may not occupy as necessarily diverse resources as their neighbor groups and thus dysfunction of some groups can arise. In this sense, random grouping might not make 'fair' groups as some argued (Bacon et al., 2001; Chapman, Meuter, Toy, & Wright, 2006). Likewise, self-selection groups are easy to be implemented too. Students are allowed to pick their desirable group members by themselves. In such cases, teachers normally do not need to put much effort into assignment. If there are 'remainders' (e.g. participants who do not have any acquaintance), teachers then have to either coordinate them to the existing groups or group them randomly. Effortlessness could be one good reason why there is a high incidence of self-selection groups in schools.

Scientific findings indicate that the self-selection method prevails over the random method because it brings about better communication and a greater degree of enthusiasm among group members (Chapman et al., 2006). Nevertheless, the shortcomings of self-selection groups are evident, one of which is the 'remainder' problem as Bacon described in (Bacon et al., 2001). Teachers can allocate remainders to the existing groups if that is the case. But it may be hard for this special cohort to get involved in group events because group cohesion has been established initially among those acquaintances. This could hurt their high motivation of participation and gradually force them to be isolated in groups. As a result, this could even put them in danger of becoming free riders or dropouts. Another drawback is over-homogeneity (Bacon et al., 2001, p. 10), which derives from humans' willingness to select team members like themselves to work with. This obviously diminishes the chance to learn from peers in such homogenous groups and may even adversely affect group performance (Bacon et al., 2001). Overall, random methods and self-selection methods may need to be applied with caution because of their aforementioned disadvantages.

2.3.2 Computer-supported grouping methods

As aforementioned in Section 2.3.1, self-selection and random methods used to be adopted in classrooms. Recently, however, such methods have increasingly been overshaded in the sense that it is hard for them to account for a variety of grouping criteria. Teachers' manual allocation can perhaps handle one single criterion in a very small class (homogeneous grouping should be easier than heterogeneous grouping if it is the case). For a larger number of students (typically in online learning contexts), considering more than one criterion, this appears to be too complex to be solved by human eyes and pencils any more (Hwang, Yin, Hwang, & Tsai, 2008). As such, computer-supported methods have been proposed as an assistant to look for an optimal or a near-optimal group formation.

Centering on computer-supported grouping methods, two research families can be recognized in terms of how they model the group formation problem. The first research family is constraint-based methods that model group formation as a constraint satisfaction problem. They normally translate the grouping criteria into constraints and then examine the violation of such constraints when they assign students into groups. Generally, the best group formation should be the one that violates the least number of the applied constraints. An example of constraints can refer to work from Ounnas et al (Ounnas et al., 2009, p. 51). They used "*Distribute Belbin roles such that every group has one leader*" as one of their sample constraints. The constraints are normally of threshold type. Thus, these constraints methods are often employed to solve the threshold-based criteria (see threshold-based criteria in Section 2.2). The constraints can also be classified into two categories, hard constraints and soft constraints (Balmaceda et al., 2014). The hard constraints are ones that cannot be violated in any case, such as, each student should only stay in one single group (assignment of one student into more than one group is not acceptable). Soft constraints can possibly be violated, but the violation will negatively affect the quality of the solution. For example, if there is a soft constraint saying that each group should have only one leader, some groups with more than one leader or no leader at all can be possibly composed, but the solution clearly does not favor those. In addition, instructors can emphasize the particular importance of some constraints resorting to a weighting strategy (Balmaceda et al., 2014). Basically, violating a more important constraint will accordingly carry a more serious penalty. These constraint-based methods are perfect to solve group formation problems of threshold-based grouping criteria.

Constraint-based methods normally need a third-party computational solver to compute the modelled constraint satisfaction problems. For instance, a DLV solver (an implementation of disjunctive logic programming) and a Choco solver were respectively leveraged in (Ounnas et al., 2009) and (Balmaceda et al., 2014). However, such solvers must limit the number of constraints, as pointed out in (Ounnas et al., 2009), because of the high computational cost. In Ounnas' work, the number was tested up to eleven (Ounnas et al., 2009). This, accordingly, limits the number of the applied grouping criteria. The constraint-based methods leave teachers much freedom to propose their own grouping criteria. The fact, however, turns out to be that many teachers are not motivated to give self-proposed grouping criteria because they have very little or even no knowledge about how to make effective groups (Srba & Bielikova, 2014).

Table 2.4 Grouping methods

Category	Grouping algorithms
Constraint-based methods	
	Logic programming (Ounnas et al., 2009)
	A Choco solver (Balmaceda et al., 2014)
Distance-computational methods	
	genetic algorithms (Bergey & King, 2014; Gogoulou et al., 2007; Hwang et al., 2008; Moreno et al., 2012)
	evolutionary algorithm (Yannibelli & Amandi, 2012a)
	ant colony optimization algorithm (Graf & Bekele, 2006)
	artificial bee colony Algorithm (C.-C. Hsu et al., 2014)
	particle swarm optimization algorithms (Lin, Huang, & Cheng, 2010; Zheng & Pinkwart, 2014)
	GroupAL algorithm (Konert, Burlak, & Steinmetz, 2014)
	fuzzy clustering algorithm (Feng et al., 2008)
	faraway-so-close algorithm (Paredes et al., 2010)
	modified rank order clustering algorithm (Srba & Bielikova, 2014)

The second family of grouping methods is distance-computational methods. Those methods make use of peer difference or similarity as a metric to measure the compatibility among group members. Since the difference and similarity are essentially represented by a numerical distance in computation, this thesis names them distance-computational methods. Using those methods, homogeneous groups and heterogeneous groups are preferably composed. For example, when composing ability homogeneous groups, the goal is basically to keep peer distance of group members' ability as close as possible. On the contrary, in heterogeneous groups, a possibly maximal distance among group members is the aim.

To solve distance-computational problems, many computer algorithms have been suggested. Amongst them, there are genetic algorithms (Bergey & King, 2014; Gogoulou et al., 2007; Hwang et al., 2008; Moreno et al., 2012), an evolutionary algorithm (Yannibelli & Amandi, 2012a), an ant colony optimization algorithm (Graf & Bekele, 2006), an artificial bee colony algorithm (C.-C. Hsu et al., 2014), particle swarm optimization algorithms (Lin et al., 2010; Zheng & Pinkwart, 2014), a GroupAL algorithm (Konert et al., 2014), a fuzzy clustering algorithm (Feng et al., 2008), a faraway-so-close algorithm (Paredes et al., 2010), and a modified rank order clustering algorithm (Srba & Bielikova, 2014) (cf. Table 2.4).

2.4 Group composition in MOOCs

In (Kizilcek, 2013), Kizilcek argued that forming learning groups might be necessary in the sense that many online MOOC students may not be autodidactic, and they could probably reap much more via group learning. Face-to-face communication is considered to be more expressive than computer-mediated communication. However, diverse viewpoints could more likely arise from geographically distributed communication rather than local groups. Therefore, he suggested a 2-step model. First, one can create in-person groups composed locally. The second step is to compose geographically-distributed groups with the global audience. This model, in principle, can reap benefits from both face-to-face communication and the diversity of a global audience.

In (Wichmann et al., 2016), Wichmann et al. conducted a study in an online Moodle[9] course about group creation. They composed heterogeneous learning groups and homogeneous groups in accordance with students' activity in an online discussion forum. The findings are twofold. First, at the low-quantity activity level, the homogenous groups yielded more productivity than the heterogeneous groups. Second, at the high/average-quantity activity level, the heterogeneous groups were more productive.

[9] https://moodle.org/

Recently, Wen from Carnegie Mellon University has made several attempts in this research field too. At NovoEd[10]'s platform, she and her colleagues experimented with small-group learning and found that leadership is an important factor in the success of MOOC groups (Wen, Yang, & Rosé, 2015). In one of her recent works, she also proposed a method to create online learning groups by using transactivity that analyzed students' interaction in an online discussion forum (Wen et al., 2016). The reported evidence shows that this method can bring about greater knowledge integration than a random assignment strategy.

2.5 Group re-composition

Throughout the long history of classroom teaching, group re-composition may sometimes occur in some learning contexts. According to a report on K-12 classrooms (Shimazoe & Aldrich, 2010, p. 55), teachers may need to recompose learning groups for several reasons. First, teachers may find dysfunction in certain groups. Second, students themselves want to dissolve their groups. In classrooms, manual redistribution of those students would probably not be a bad choice. That could be the reason why group re-composition happens in some cases, but has not been systematically studied yet.

When moving to online learning, manual reassignment of students into new groups might not be actionable. The student audience is too big, diverse and remote. Teachers may not be able to solve this group re-composition in an efficient way by their hands. Thus, automating the whole process is interesting to be explored. It appears that only one recent publication attempts to address this issue (Srba & Bielikova, 2014). The authors proposed a dynamic group formation method to improve Computer Supported Collaborative Learning (CSCL) groups. They retrieved group interaction data and iteratively made use of it to compose groups for each new task in their course. This work pioneers the use of group dynamics to compose groups. Yet, it still leaves much room for improvement. First, the method relies on the pre-defined grouping criteria that may be questionable for any specific application scenario. Second, it is necessary to scale up the method for larger online learning platforms.

[10] http://novoed.com/

2.6 Chapter summary and reflections

This chapter goes through the relevant literature about the group composition and re-composition topic. The research of group composition dates back to decades ago. The random methods and self-selection methods are often employed by many grouping practitioners owing to their ease of implementation. Recently, as groups have been empirically studied within a variety of settings (e.g. CSCL learning groups and project-based teams), many factors have been found to matter to group success (e.g. group members' personalities, ability level and team roles). Moreover, many grouping criteria have been tested as well, such as, learning styles heterogeneous groups. Due to the emergence of those factors and grouping criteria, the group composition problem becomes more complex than ever. Random methods and self-selection methods are too criteria-blind to account for those grouping criteria. Any manual method cannot solve this problem very efficiently either especially when the student enrollment grows to be very big, like in recent MOOCs. Regarding efficiency, Computer-supported methods (i.e. computer algorithms) would be a better choice.

Thus far, plenty of algorithms have been proposed to solve the group composition problem. Nevertheless, the lack of comparison work on those existing grouping algorithms stresses many grouping practitioners – there are quite many algorithms. Those grouping practitioners may often look for guidance that can tell them which algorithm(s) is better than others in their specific application scenarios. Additionally, with the advent of larger and larger learning environments, testing those existing group composition methods on a bigger data set becomes necessary as well.

Recently, some researchers have begun to compose small learning group on newly-formed learning platforms (i.e. MOOCs). Despite the difference among their grouping methods, they do commonly realize that small learning groups can bring about pedagogic benefits and different learning experiences to online students. We can collect students' geographic information and compose local groups as Kizilcek suggested (Kizilcek, 2013). We can also create MOOC heterogeneous or homogeneous groups according to insights derived from students' interaction in a course forum (Wichmann et al., 2016). More recently, Wen et al. made use of students' interaction data and proposed a machine learning method to assign students into suitable groups (Wen et al., 2016). In this realm of research, although small, the trend is evident: composing learning groups with machine learning methods and students' interaction data takes place and could be the future.

Besides, group composition in MOOCs faces new challenges. The widely reported high drop-out rate in current MOOCs could result in group instability, that is, many students drop out and thus cause many groups incomplete in size. If they still need to pursue their group work, the left-behind human resource would be far too short-handed to achieve the intended group goals. In such foreseeable cases, group re-composition would mitigate this problem via re-assigning students from those risky groups into new groups. The group re-composition topic is very new and has been rarely studied thus far.

3 GROUP COMPOSITION

Composing small learning groups could be much easier if we adopt random methods and self-selection methods. This is the reason why random groups and self-selection groups are still very popular in current schools. However, these methods are not able to account for any grouping criterion. For example, composing ability heterogeneous groups via these methods would be wrong. Rather, computer algorithms can help us to meet such grouping criteria. So far, many computer algorithms have been proposed to solve the group composition problem, but little work has been done to compare those algorithms. In addition, applying those algorithms to a MOOC is also very new.

This chapter will answer the research question RQ1 by looking into four selected grouping algorithms and comparing their performance in grouping quality and time cost. Besides, a MOOC experiment will tell us how to compose learning groups in current MOOCs and the impact of learning groups on drop-out and learning performance, which seeks to address the research question RQ2.

This chapter starts with mathematical modelling of the group composition problem. Next, it elaborates the approaches to solve this problem. Two simulation experiments are then described to compare those selected grouping algorithms. It ends with a MOOC experiment in an attempt to validate grouping algorithms in a real learning environment and observe the impact of small learning groups.

3.1 Modelling the Group Formation Problem

The group formation problem is essentially a combinatorial problem. Assume $P = \{p_1, p_2, \dots, p_n\}$ is a population of students, each p stands for a student, n denotes the total amount of students. The task of group formation is to assign those students into small learning groups with k students in each. Here $k \leq \left\lceil \frac{n}{2} \right\rceil$, which constrains that the whole population has to be segmented at least into two groups. Besides, the resulting groups need to satisfy the given grouping criteria.

Grouping criteria, in this thesis, are composed of a set of aforementioned grouping attributes (e.g. personalities) and their distribution over each learning group (e.g. homogeneously, heterogeneously). For instance, "*group students with similar background knowledge*" could be a criterion, which indicates that we need to make the attribute, *background knowledge*, inside each group as homogeneous as possible. Those criteria could also be threshold-based, for example, in (Balmaceda et al., 2014), the number of group members who play the same role was limited to up to half of the group size. As mentioned in Section 2.3.2, these criteria can be perfectly met by the constraint-based methods. However, those methods are reported to rely on third-party solvers and limit the number of criteria because of computational efficiency. This thesis seeks to model and solve the group composition problem using a distance-computational method rather than a constraint-based method, which implies that the modelling is a good fit to heterogeneous groups and homogeneous groups.

Let $G = \{p_1, p_2, \dots, p_k\}$ be a group of students. Assume a group has k students in total ($k \in Z^{+} and\ k > 1$). p_i denotes the i_{th} student in the group G, $p_i \in G$. Each student has a tuple of attributes $A = (a_{i1}, a_{i2}, \dots\ a_{il})$ that represent his/her characteristics in a numerical manner. For example, a_{ij} could be learning performance. Here the value of each attribute has preferably to be coded into a numerical value and standardized in the range of 0 and 1. Where a_{ij} is the value of the i_{th} student's j_{th} attribute, $a_{ij} \in A$. l is the total number of attributes to be incorporated, and $i \leq k$, $j \leq l$. Each attribute should take its own importance into account. Let w_m stand for the m_{th} attribute's weight, $w_m > 0$. We define the interpersonal compatibility of a pair of students as an Euclidian distance between two students, which is denoted by CP_{ih}, $h \leq k$, $i \neq h$. It can thus be calculated in Equation (3-1).

$$\boldsymbol{CP_{ih}} = \sqrt{\frac{\sum_{m=1}^{l} w_m (a_{im} - a_{hm})^2}{\sum_{m=1}^{l} w_m}} \qquad (3\text{-}1)$$

We next define one's compatibility to the whole group as an average of his/her compatibility to all other members, which is signified by PCP_i and computed in Equation (3-2).

$$\boldsymbol{PCP_i = \frac{1}{k-1}\sum_{h \neq i} CP_{ih}} \tag{3-2}$$

Where $i \leq k$ and $h \leq k$. Then the quality of a resulting group can be examined by an average of all of its members' compatibility, as shown in Equation (3-3).

$$\boldsymbol{GCP = \frac{1}{k}\sum_{i=1}^{k} PCP_i} \tag{3-3}$$

For example, Smith, Eva and Bob are three students. If we compose these three into a group according to their knowledge level and motivation level, assume their data is given as shown in Table 3.1 and knowledge level and motivation are of equal importance (with 0.5 to each). According to Equation (3-1), we can calculate $\boldsymbol{CP_{Smith,Eva}} = \sqrt{\frac{0.5\times(0.9-0.5)^2+0.5\times(0.5-0.7)^2}{0.5+0.5}} = 0.316$. Similarly, we can calculate $\boldsymbol{CP_{Smith,Bob}} = 0.474$. Next, we can calculate Smith's compatibility to the group according to Equation (3-2), $\boldsymbol{PCP_{Smith}} = 0.395$. Applying the same laws, we can then calculate $\boldsymbol{PCP_{Eva}} = 0.349$ and $\boldsymbol{PCP_{Bob}} = 0.428$ respectively. Finally, we use Equation (3-3) to calculate the whole group's grouping quality, $\boldsymbol{GCP} = (\boldsymbol{PCP_{Smith}} + \boldsymbol{PCP_{Eva}} + \boldsymbol{PCP_{Bob}}) \div 3 = 0.391$.

Table 3.1 Example data for three students

Name	**Knowledge level**	**Motivation level**
Smith	0.9	0.5
Eva	0.5	0.7
Bob	0.3	0.2

The quality of a group formation, in this thesis, is defined as the average quality of all composed groups. Assume M groups are composed totally, $M = n/k$, and GCP stands for a group's quality (given in Equation (3-3)), then the average quality of a whole group formation can be calculated as Equation (3-4).

$$GFQ = \frac{\sum_{i=1}^{M} GCP_i}{M} \quad (3\text{-}4)$$

In reality, one's characteristics are not always the case of numerical data. They might be nominal data (categorical data without an intrinsic order, e.g. team roles), ordinal data (categorical data with an intrinsic order, e.g. motivation level measured with *low*, *medium* and *high*) and dichotomous data (e.g. gender). When applying Equation (3-1) (3-2) and (3-3), the necessity comes to numerical coding of such data. With regard to nominal data, a binary one-hot (aka one-of-K) coding can be applied (see details in scikit-learn online documentation[11]), which extends the existing attributes set with all possible values of the nominal attributes. As an example in Figure 3.1, one can extend the nominal attribute *language* with its values (i.e. German, English and Chinese etc.). With each newly extended attributes, one can instead use a numerical value, either 0 or 1, to symbolize. For instance, ('language': *English*) can be accordingly transformed into ('language=English': 1 , 'language=German': 0 , 'language=Chinese': 0). For ordinal data, the coding must be processed with consideration of the values' intrinsic order instead. If an attribute has values *low*, *medium* and *high*, the inherent ascending or descending tendency has to be numerically encoded (or mapped) accordingly. Often, dichotomous data can be simply replaced by either 1 or 0 in many cases.

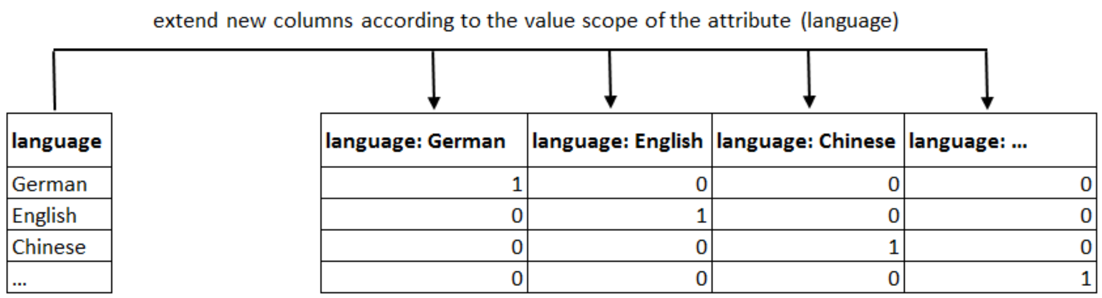


Figure 3.1 One-hot coding example

Another problem is how to deal with homogenous attributes and heterogeneous attributes or a combination of both. Regarding heterogeneity of each pair of students, we can simply use their interpersonal compatibility, as shown in Equation (3-1), to

[11] http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html

represent it, since this interpersonal compatibility is a Euclidian distance between a pair of students. To make it more clear, we can re-write Equation (3-1) as shown in Equation (3-5), where $w_m > 0$. Then we know that we must keep such compatibility as high as possible when we compose heterogeneous groups.

$$\boldsymbol{CP_{ih}}^{\boldsymbol{hete}} = \sqrt{\frac{\sum_{m=1}^{l} w_m (a_{im} - a_{hm})^2}{\sum_{m=1}^{l} w_m}} \qquad (3\text{-}5)$$

However, for homogeneous attributes, if we keep that compatibility high, that would lead us in the wrong direction. In such cases, we can compute it as Equation (3-6), such that we can still make that compatibility as high as possible, but it goes into a direction of homogeneity.

$$\boldsymbol{CP_{ih}}^{\boldsymbol{homo}} = \mathbf{1} - \sqrt{\frac{\sum_{m=1}^{l} w_m (a_{im} - a_{hm})^2}{\sum_{m=1}^{l} w_m}} \qquad (3\text{-}6)$$

In the case of a mix of heterogeneous and homogeneous attributes, the compatibility can be computed as Equation (3-7).

$$\boldsymbol{CP_{ih}}^{\boldsymbol{mix}} = \frac{w_{hete} \times CP_{ih}^{hete} + w_{homo} \times CP_{ih}^{homo}}{w_{hete} + w_{homo}} \qquad (3\text{-}7)$$

Where $\mathrm{w_{hete}}$ and w_{homo} are weighting importance of heterogeneous and homogenous attributes respectively ($\mathrm{w_{hete}} > 0$ and $w_{homo} > 0$).

3.2 Approaches to the group formation problem

3.2.1 Exact method

Since there are many possibilities of assigning students into groups, one student could be assigned into two different groups by two different teachers. To seek the best group formation, an exact yet naïve approach is to enumerate all possible combinations and then pick the best solution among them. Assume $n \bmod k == 0$, which means the whole set of n students is evenly assigned into n/k groups. We first select k students from the set, which results in $\binom{n}{k}$ possibilities. For selection of the next k students, we will have $\binom{n-k}{k}$ possibilities. So continues this combination until the last k students are left to compose the last group. The number of above possibilities can be calculated as Equation (3-8).

$$\prod_{i=0}^{\frac{n}{k}-1}\binom{n-ik}{k}=\frac{n!}{(n-k)!\,k!}\times\frac{(n-k)!}{(n-k-k)!\,k!}\times\cdots\times\frac{k!}{0!\,k!}=\frac{n!}{(k!)^{n/k}} \tag{3-8}$$

Among these possibilities, there are some repetitive combinations. For instance, 9 students, $\{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9\}$, can be composed into three groups, $[(p_1, p_2, p_3), (p_3, p_4, p_5), (p_7, p_8, p_9)]$, if the group size is set to three. Following the above law, it can also be composed as $[(p_1, p_2, p_3), (p_7, p_8, p_9), (p_3, p_4, p_5)]$. However, these two group formations are, in fact, same. We thus need to remove such repetitions by a factor of $P\left(\frac{n}{k},\frac{n}{k}\right)=(\frac{n}{k})!$ besides. The exact number of all possible group formations can be computed as Equation (3-9).

$$\frac{\prod_{i=0}^{\frac{n}{k}-1}\binom{n-ik}{k}}{P\left(\frac{n}{k},\frac{n}{k}\right)}=\frac{\frac{n!}{(k!)^{\frac{n}{k}}}}{\left(\frac{n}{k}\right)!}=\frac{n!}{(n/k)!\,(k!)^{n/k}} \tag{3-9}$$

As shown in Equation (3-9), if $k = 1$ or $k = n$, the problem can be solved in $O(1)$, because there is only an unique group formation in both situations. Otherwise, the problem was considered as a NP-hard problem in some work (Lin et al., 2010; Yannibelli & Amandi, 2012a), which means that the number of possible combinations increases exponentially as n grows (Woeginger, 2003). This certainly makes a solution to such a problem computationally expensive. In big classes, especially in up-to-date MOOC courses with thousands of students, this exact method can hardly solve the group composition problem in an efficient way.

3.2.2 Heuristic methods

Since the optimal solution to the group composition problem is expensive to be found (since the time cost grows exponentially as the number of students increases, see details in Section 3.2.1), a near-optimal solution rather than the optimal one could be a realistic alternative in many real-life cases. A heuristic is often used to gain that near-optimal solution. It starts with a bunch of initial solutions (could be random solutions), then selects the best available solution from the current availability, and then moves all solutions to the so-far best, which could yield a new and better solution. It runs the upper two steps iteratively, which always seeks the ‘best’ and accordingly updates solutions iteration by iteration. Note that some heuristics could start with one single solution rather than multiple ones, and then optimize the solution by internally exchanging positions of some elements, which is often called local heuristics. A heuristic search consists of a fitness function and an approximation algorithm. The fitness function is used to evaluate newly-produced solutions. The algorithm controls the whole process of search via maximizing or minimizing the output value of the fitness function.

Let the quality of a group formation, GFQ, indicate the fitness value, then

$$\boldsymbol{f = GFQ = \frac{\sum_{i=1}^{M} GCP_i}{M}} \qquad (3\text{-}10)$$

In this thesis, the objective of the heuristics is to maximize the fitness value, f, in order to best satisfy the grouping criteria, as shown in Formula (3-11).

$$\boldsymbol{Maximize\left(f = \frac{\sum_{i=1}^{M} GCP_i}{M}\right)} \qquad (3\text{-}11)$$

Thus far, there have been many heuristics to solve the group composition problem (cf. Section 2.3.2), for instance, genetic algorithms (Bergey & King, 2014; Gogoulou et al., 2007; Hwang et al., 2008; Moreno et al., 2012), an artificial bee colony algorithm (C.-C. Hsu et al., 2014), an evolutionary algorithm (Yannibelli & Amandi, 2012a) and an ant colony optimization algorithm (Graf & Bekele, 2006). However, As a survey reports in (Cruz & Isotani, 2014), only 2% of those algorithms provide source code. Due to this reason, only four algorithms turned out to be selected in this thesis, namely, a discrete-PSO algorithm, a genetic algorithm, an ant colony optimization and an adapted k-means clustering algorithm. The following starts to elaborate the selected four algorithms in detail.

3.2.2.1 A discrete-PSO algorithm

Particle swarm optimization (PSO), inspired from bird flocking and fish schooling, was first proposed by Eberhart and Kennedy to optimize nonlinear functions (Eberhart & Kennedy, 1995). PSO searches the problem space and looks for the optimal solution by updating particles which were randomly generated at the initial stage of the algorithm. These particles have the ability to memorize their own best positions in the search space and to share the best position among a group of particles (namely, a swarm). Particles move and adjust their positions iteratively according to their personal best prior positions and the global best position among the swarm. PSO was originally developed to solve optimization problems in continuous space. Subsequently, researchers have developed variants of the standard PSO to solve discrete problems, such as the manufacturing cell design problem (Duran, Rodriguez, & Consalter, 2008) and the travelling salesman problem (Changsheng, Jigui, Yan, & Qingyun, 2007). So far, the use of discrete-PSO to solve the group composition problem has rarely been reported.

As shown in Figure 3.2, n students are simply permutated in a list. Assume that the group size is k, students in positions of $(1, 2, \ldots k)$ stay in a group, and another k students in positions of $(k+1, k+2, \ldots 2k)$ stay in the second group, and $(i \times k + 1, i \times k + 2, \ldots i \times k + k)$ is the i_{th} group. This is a typical example of a group formation. Note that if we change the permutation of these n students, we can get another group formation. The discrete-PSO starts with a given number of such group formations. Each group formation is called a particle in discrete-PSO algorithm. The goal of this algorithm is to update these initial group formations and searches for the optimal group formation iteratively. In process of the iterative searching, a velocity vector v_k^{t+1} is required to update a particle P_k into its new generation for the next iteration.

Position →	1	2	3	4	…	n-3	n-2	n-1	n
Students →	S_1	S_2	S_3	S_4	…	S_{n-3}	S_{n-2}	S_{n-1}	S_n

Figure 3.2 Students' representation in the discrete-PSO algorithm

discrete-PSO algorithm pseudo-code

```
Parameters:
        - the number of particles (N)
        - the number of iterations (nIter)
        - inertial weight (ω)
        - self-learning factor (c_1)
        - social learning factor (c_2)
Steps:
        randomly initialize a population of N particles;
        i = 0;
        do{
                calculate the fitness of each particle using Equation 3-10;
                for each particle
                        update the personal best and global best;
                        update velocity using Equation 3-12;
                        update the particle using Equation 3-13;
                i += 1;
        }(while i < nIter)
         output the global best;
```

Equation (3-12) is used to calculate v_k^{t+1}. t denotes the t_{th} iteration and k indicates the k_{th}particle. P_k^t, and $PE_{k,best}$ are the current state and the personal best prior state of P_k respectively. G_{best} is the global best particle. ω, c_1, c_2 are learning coefficients. v_k^t is the velocity vector of the prior iteration. Based on the equation, the up-to-date velocity is not merely determined by the differences from the current to the global best and to the personal best, but also it follows the prior velocity (v_k^t)'s trend to some extent.

$$\boldsymbol{v_k^{t+1} = \omega * v_k^t + c_1(PE_{k,best} - P_k^t) + c_2(G_{best} - P_k^t)} \qquad (3\text{-}12)$$

A velocity vector is essentially a set of pairwise sequences. As shown in Figure 3.3, assume $v_k^{t+1} = [(2,4),(6,8)]$ and $P_k^t = [s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9]$. Then we can update P_k^t into its next generation $P_k^{t+1} = [s_1, s_4, s_3, s_2, s_5, s_8, s_7, s_6, s_9]$ using Equation (3-13). Note that the initial velocity v_k^0 can either be an empty list or a random list of moving sequences.

$$\boldsymbol{P_k^{t+1} = P_k^t + v_k^{t+1}} \qquad (3\text{-}13)$$

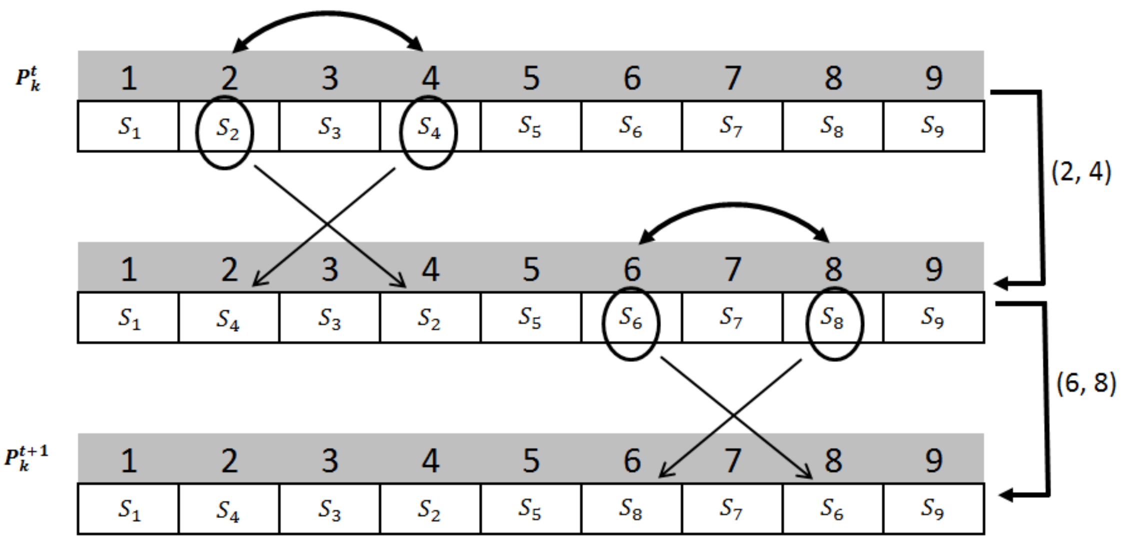


Figure 3.3 Velocity to update particles

With respect to the fitness value, the global best particle is a group formation that is superior to all others over the past iteration(s). Implicitly, all particles share this global best information. Apart from the global best, each particle needs to remember its own best group formation alongside the history of its evolvement also, namely the personal best. Conversely, this personal best is not shared with all other particles.

With regard to Equation (3-12), many intricacies need to be explained. First, what is the subtraction operator of two particles? Assume $P_1 = [s_5, s_4, s_2, s_9, s_3, s_7, s_1, s_6, s_8]$ and $P_2 = [s_5, s_9, s_2, s_4, s_6, s_7, s_1, s_3, s_8]$. As shown in Figure 3.4, first of all, we need to compare P_1 and P_2 from the beginning to the end. When we find the first difference at position 2, we then try to find s_9 in P_1 and exchange it with the one at position 2 aiming at making this position same. We then continuously find the second difference at position 5 and exchange it with the one at position 8 in a same fashion. The result turns out to be a set of moving sequences that can move P_1 to P_2. We can consider the subtraction operator as a reverse operation of the addition operator in Equation (3-13).

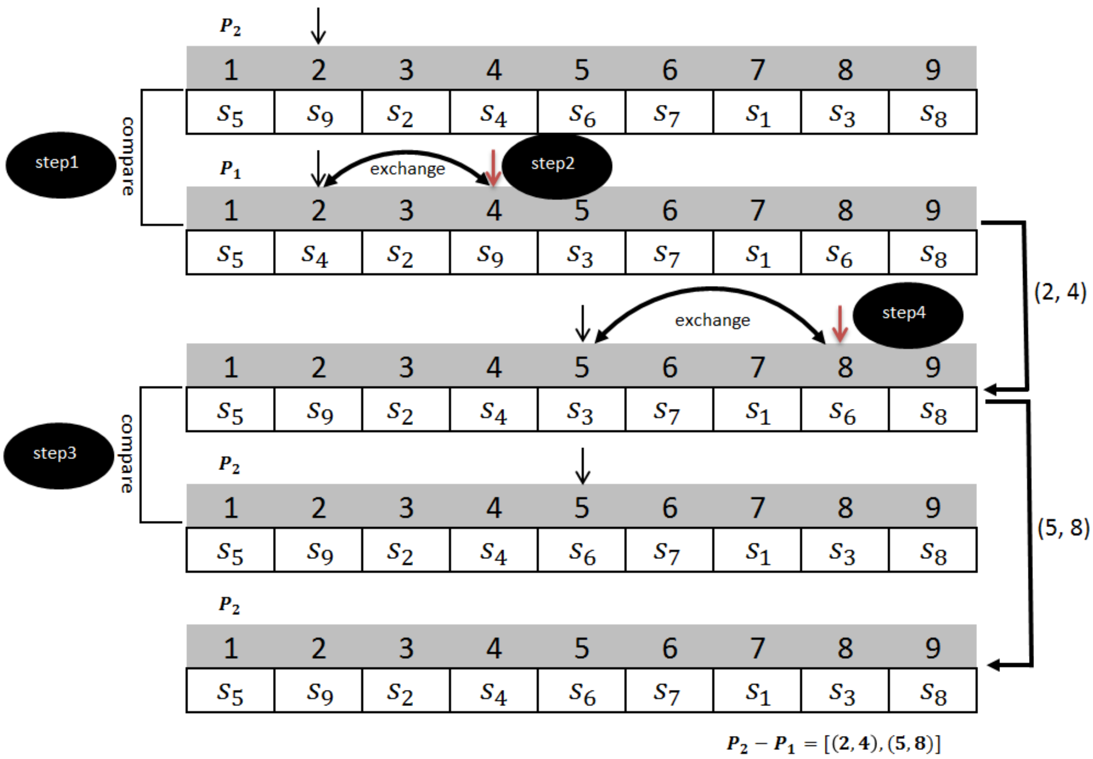


Figure 3.4 Subtraction of two particles

ω is a inertial weight. This coefficient indicates to what extend the particles maintain their current movement direction. c_1 is a self-learning factor that represents how much a particle moves towards the personal best position. c_2 is a social-learning factor, which means how much a particle moves to the best experience obtained by the whole swarm of particles. These three factors together control the movement of each particle with respect to a balance among the current moving direction, the personal best and the global best positions.

The multiplication operator, such as $\omega * v_k^t$, means how many moving sequences should be taken from the velocity vector. Assume that $\omega = 0.5$ and $v_k^t = [(2,4),(5,8)]$, then 50% of the velocity vector v_k^t will be randomly taken, as shown in Equation (3-14) (since it is a random selection, the result could also be $[(5,8)]$ in some cases).

$$\boldsymbol{\omega} * \boldsymbol{v_k^t} = \mathbf{0.5} * [(\mathbf{2},\mathbf{4}),(\mathbf{5},\mathbf{8})] = [(\mathbf{2},\mathbf{4})] \tag{3-14}$$

Lastly, a sum of two velocity vectors is to combine two vectors into one vector. For instance, $[(\mathbf{2},\mathbf{4})] + [(\mathbf{5},\mathbf{8})] = [(\mathbf{2},\mathbf{4}),(\mathbf{5},\mathbf{8})]$.

3.2.2.2 A genetic algorithm

To solve the group composition problem, much attention has been paid to the adoption of genetic algorithms (GA) (Hwang et al., 2008; Jozan, Taghiyareh, & Faili, 2012; Moreno et al., 2012; Yannibelli & Amandi, 2012b). Based on relevant literature (Reeves, 1995; Safe, Carballido, Ponzoni, & Brignole, 2004; Whitley, 1994), the genetic algorithm can be generally summarized as follows. It generally starts with an initial population that consists of a certain number of solutions to the given problem. In the group composition problem, those solutions can be a list of all students. Besides, additional information needs to be encoded in such solutions. For instance, from the beginning of the list, every *M* students belong to a group (if the group size is *M*). Additionally, a fitness function needs to be used to evaluate such solutions in order to reflect the quality of the generated group formations. In this thesis, Equation (3-10) can properly function this. Next, GA attempts to improve these initial solutions by evolving them into new generations. This procedure is analogous to the biological evolution. It consists of three sequential operations. First, it selects the most suitable individuals (i.e. parents) out of the population so that their good genes can be preserved for the next generations. Second, those parents' chromosomes (i.e. solutions to a given problem) need to crossover in order to keep both parents' good gene. Third, some mutations are allowed to occur. These three operations are, in short, named as *selection*, *crossover* and *mutation*. They will be described in detail in the following sections. When the algorithm satisfies pre-defined criteria, it terminates and outputs the best solution among the offspring. Figure 3.5 shows the general workflow of the genetic algorithm.

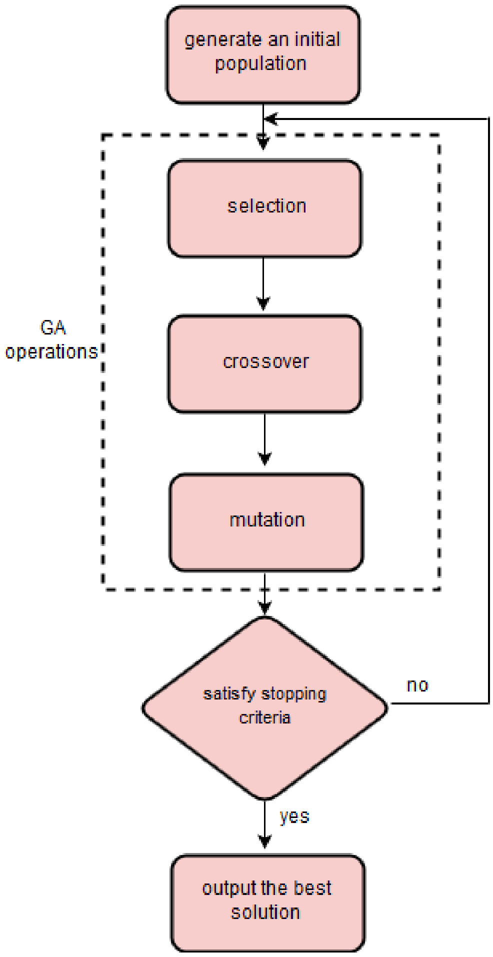


Figure 3.5 Paradigm of the genetic algorithm

Selection: this thesis applies a 2-tournament selection strategy that has been previously advised in (Agosten E Eiben & Smith, 2010; Yannibelli & Amandi, 2012b). Basically, every parent candidate is the winner of a pair of randomly chosen chromosomes from the population. One will win if his fitness value is superior to the other.

Crossover: crossover is controlled by a parameter called crossover rate (P_c). If a randomly generated number ($rn \in [0, 1]$) is less than the cross rate, it performs the operation of crossover. Otherwise, it simply leaves the selected pair of parents alone. A two-point crossover policy was employed in this thesis. As shown in Figure 3.6, first, it randomly generates two indices, $index_1$ and $index_2$. Second, $parent_1$ descends his gene between $index_1$ and $index_2$ to $child_1$. Similarly, $parent_2$ descends his gene between $index_1$ and $index_2$ to $child_2$. Third, from $index_2$ onwards, $parent_2$ begins to descends his gene to $child_1$, if the selected gene is already between $index_1$ and $index_2$, it skips that. When it goes to the end of the data structure, it starts from the beginning of it until every gene is filled. A similar moving pattern applies to fill $child_2$ also. In the end, as we can see, $child_1$ and $child_2$ inherit the gene from both parents.

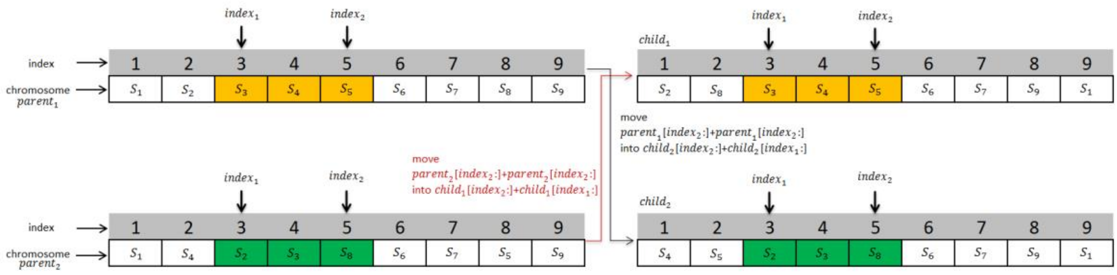


Figure 3.6 Crossover operation

Mutation: like crossover, mutation also occurs probabilistically. As long as a randomly generated number ($rn \in [0, 1]$) is less than the mutation rate (P_m), it performs the operation of mutation. It first randomly selects two positions and then exchanges the elements in the selected positions (cf. Figure 3.7). Such a slight change modifies the sequence of the list and it turns out to make a different group formation. Note that, if the selected two positions happen to be encoded in one group, it then does not actually make any change of the group formation. Nevertheless, it occurs very rarely in principle.

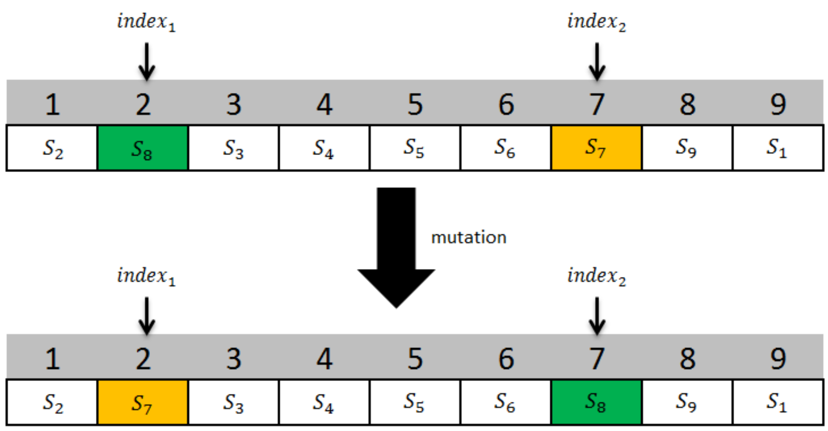


Figure 3.7 Mutation operation

3.2.2.3 An adapted k-means clustering algorithm

The adapted k-means clustering algorithm mentioned in this thesis was proposed by Dirk Uys from P2PU[12] community. His original work was trying to group students

---

[12] https://www.p2pu.org/

with similar musical preference (P2PU, 2014). As shown in the following pseudo-code, the adapted k-means clustering algorithm runs a local search procedure iteratively. In each iteration, it exchanges students from different groups in attempts to possibly improve the average score (a measure of the group quality) of the entire group formation.

Adapted k-means clustering algorithm pseudo-code (P2PU, 2014)

```
step 1:
randomly assign users to groups;
step 2:
for every group:
    for every user in the current group:
        calculate the possible group scores;
         for the user in all the other groups:
         if the user has a higher group score in one of the other groups:
             find the user in the other group with the lowest group score;
             swap the two users;
step 3:
while we are significantly improving the average group score, go back to step 2
```

3.2.2.4 Ant colony optimization

Ant Colony Optimization (ACO) was proposed in the 1990s and aims to efficiently address the combinatorial problems (e.g. the well-known Traveling Salesman Problem (TSP)) (Dorigo & Gambardella, 1997). Afterwards, for the sake of the original Ant System (AS) algorithm’s lower competition than the other algorithms for TSP in those days, many other ants-based algorithms have been proposed, typically, MAX-MIN Ant System (MMAS) and Ant Colony System (ACS) (Dorigo & Stützle, 2009). The ACO simulates the ant colony’s behavior in nature. Ants lay pheromone along the paths when they search for food source. Their peers sense the pheromone, follow the paths and continuously accumulate on the trail. Through sharing this collective information (pheromone), they are finally able to find shortest paths to the food source. Reflecting to specific applications, the paths that ants construct are solutions to a given problem (e.g. paths of ants can be tours of salesman in TSP). The quality of these solutions is measured by a fitness function. When an ant decides to follow a path, it prefers to choose a better path (depends on the distance and the strength of the pheromone). Given a certain amount of iterations, all artificial ants finally opt for walking on the best path (analogous to ants’ moves that we observe in nature).

Recently, Graf et al. applied ACS to solve the learning group composition problem (Graf & Bekele, 2006). In their work, they modelled the group composition problem as a graph. The graph is made up of *n* nodes which represent *n* students. The artificial ants pass through all these nodes but only with one visit to each node. Simultaneously, they lay pheromone on the edges between each pair of nodes as they travel along. The entire tour of an ant can then be decoded as a group formation. Specifically, the first m (i.e. group size) nodes (i.e. students) are the first group and the next m ones are the second group and so on. In this model, the ants move to the next node with preference of incorporating a student that can make the current group as heterogeneous as possible. When the ants move to a node that is decoded to the first member of a new group, they need to randomly select this move, because there is no pivot to compute the heterogeneity of a new group with only one single student. Once all ants finish their tours, global information should be placed on the globally best tour in order to increase the probability for ants to choose this tour in the next iteration. This thesis replicates such same modeling as in (Graf & Bekele, 2006).

Ant colony optimization pseudo-code

```
for each pair of students (S_i,S_j),i≠j:
        initialize τ(S_i, S_j )=τ_0;
while (!stop condition)
        for each ant:
                randomly assign a starting student;
        for each of the remaining N-1 students:
                for each ant:
                        if the order of the next student is the first one in a new group:
                                randomly choose a student;
                        else:
                                choose a student according to the state transition rule;
                                locally update the pheromone;
        for each ant:
                apply the 2-opt local search to the group formation;
        globally update the pheromone;

output the best group composition;
```

As shown in the ACS algorithm, $\boldsymbol{\tau_0}$ is the initial amount of pheromone which is a constant, i.e., $\tau_0 = \frac{1}{n} \times H_{nn}$ where n is the number of ants and H_{nn} is the group formation's heterogeneity produced by the nearest neighbor heuristic (Dorigo & Gambardella, 1997). **The state transition rule** determines the selection of the next member of the current group. Literally, it follows the exploitation and biased exploration policy controlled by a constant q_0 ($0 \le q_0 \le 1$), when a randomly generated parameter q is less than q_0, the state transition rule favors the student who brings the highest heterogeneity and most pheromone. Otherwise, it selects a student by applying the random-proportional rule (Colorni, Dorigo, & Maniezzo, 1991, 1992; Dorigo & Gambardella, 1997; Dorigo, Maniezzo, & Colorni, 1996). **Locally update the pheromone** is used to update ants' pheromone on every move. For instance, it updates the pheromone as Equation (3-15) when an ant travels from the current node r to the next node s,

$$\boldsymbol{\tau(r,s) = (1-\rho)\cdot\tau(r,s) + \rho\cdot\Delta\tau(r,s)} \qquad (3\text{-}15)$$

where ρ is the local pheromone decay parameter$(0 < \rho < 1)$, and $\Delta\tau(r, s)$ is assigned to be equal to τ_0 hereby according to Dorigo's experiments (Dorigo & Gambardella, 1997). When all ants complete their tours, a local search strategy is necessarily applied. To be simple, ACS attempts to improve solutions iteration by iteration. If we can adopt a local optimization policy for those intermediate solutions before each new iteration starts, the ACS would more effectively achieve the optimal solution. In this thesis, **2-opt** was used to search for the local optimum (Croes, 1958; Graf & Bekele, 2006; Hoos & Stützle, 2005; D. Johnson, 1990; D. S. Johnson & McGeoch, 1997). **Globally update the pheromone** only allows the ant with the globally best tour (i.e. the best solution to a given problem) to update the pheromone along the whole route with a purpose of highlighting the best solution for next iterations (Dorigo & Gambardella, 1997). The globally updating follows the rule in Equation (3-16).

$$\boldsymbol{\tau(r, s) = (1 - \alpha) \cdot \tau(r, s) + \alpha \cdot \Delta\tau(r, s)} \qquad (3\text{-}16)$$

Similarly, α is a parameter that controls globally updating pheromone $(0 < \alpha < 1)$. The amount of updating pheromone $\Delta\tau(r, s)$ is defined as the heterogeneity of the globally best group formation, which is, however, different from Equation (3-15).

3.2.2.5 Methodological differences

Up to now, the selected four algorithms have been elaborated. One thing is common to all of them: they need an optimization process. This optimization process, in general, means that they cannot give a good solution to the group composition problem at the very beginning, but rather they can improve that solution iteration by iteration. Another thing is also clear: they apply different strategies to improve the solution. Borrowing some concepts from Dorigo and Gambardella's work (Dorigo & Gambardella, 1997), we can categorize the heuristics into two classes: *tour constructive heuristics* and *tour improvement heuristics*. Tour constructive heuristics basically apply the heuristic approach when they construct the solution. For example, the mentioned ant colony optimization algorithm is a typical tour constructive heuristic. It starts with one random student and then assigns the students who best meet the grouping criteria to a group. Assignment of each student is one time of applying the heuristic approach. On the contrary, tour improvement heuristics do not apply the heuristic approach until an initial solution is ready. In this thesis, the discrete-PSO algorithm, the genetic algorithm and the adapted k-means clustering algorithm belong to this category. They start with a random group formation (an initial solution to the group formation problem) and then make use of strategies to improve that group formation until they meet a termination condition.

For heterogeneous criteria or homogeneous criteria, when assigning a student to a group, the tour constructive heuristics can apply a greedy algorithm that makes the selected student's distance to the existing group members either as far as possible (heterogeneous) or as close as possible (homogeneous). But for some other criteria, such as "the number of coordinators in a group should not be more than half of the group size" and "at most one third group members should show introvert personality traits", the assignment of each student might be more difficult if we cannot see it at a group level but merely using peer distance to judge. Some of the tour improvement heuristics, on the other hand, are immune to such criteria. Because, as aforementioned, those tour improvement heuristics start with an initial group formation. With that initial group formation, one can easily calculate how many groups satisfy those criteria and thus know how good the grouping quality is. Though in a same category, the adapted k-means clustering algorithm is different than the discrete-PSO algorithm and genetic algorithm. The adapted k-means clustering algorithm starts with an initial solution, which seems to be immune to the aforementioned criteria, but it applies a local heuristic approach to improve the initial group formation. This local heuristic approach factually gauges the peer distance so as to decide which pair of students should exchange from their current groups. Thus, the adapted k-means clustering algorithm has difficulties coping with those criteria. Differing from the adapted k-means clustering algorithm, the discrete-PSO and genetic algorithm do not rely on the peer distance to improve the initial solution(s). Rather, they consider a solution as a sequence of students and analyze the differences among those initial sequences. Their goal is to move the sequences towards the so-far best. As we can see, along the whole process, they always account for the grouping criteria at a group level. Hence, in principle, such algorithms are actable to any sort of grouping criterion.

All in all, the discrete-PSO and genetic algorithm can be employed to cope with any type of grouping criteria. However, the ant colony optimization algorithm and adapted k-means clustering algorithms merely fit to the heterogeneous and homogeneous grouping criteria. For other grouping criteria, they might have difficulties and need to re-model the group composition problem.

3.3 Simulation Experiments

This section will focus on two simulation experiments. First, it will compare the discrete-PSO algorithm to an exact method and a random method from two aspects: computational performance and stability. The discrete-PSO algorithm was proposed by the author (Zheng & Pinkwart, 2014). Via this simulation, we can examine its computational performance and stability. Second, this section will compare the four selected heuristic algorithms (i.e. a discrete-PSO algorithm, a genetic algorithm, a k-means variant algorithm and an ant colony optimization algorithm) using a MOOC data set. The results are hypothesized to answer the first research question (RQ1) in Section 1.2.

3.3.1 Simulation experiment I: discrete-PSO vs exact methods and random methods

3.3.1.1 Data

Gender and MBTI personality types were taken as the grouping attributes. The selected two attributes have been widely used in the previous work, especially MBTI personality types (Balmaceda et al., 2014; Bergey & King, 2014; Srba & Bielikova, 2014). The MBTI personality was developed by Katharine Cook Briggs and her daughter, Isabel Briggs Myers based on Carl Gustav Jung's psychological theories. MBTI personality covers four dichotomies: Extrovert versus Introvert (E/I), Sensing versus Intuitive (S/N), Thinking versus Feeling (T/F), and Judgment versus Perception (J/P) (Rutherfoord, 2001; White, 1984). Composing these four dichotomies can produce sixteen different MBTI personality types (e.g. ESFP and ISFJ). As the aforementioned works did, here this thesis composes groups with balanced gender and heterogeneous personalities too.

For this simulation, 8 random data sets were generated varying data size from 9 to 3000 students. As a data sample shown in Table 3.2, $Gender = 1$ indicates a male students. $Gender = 0$, on the contrary, indicates a female student. $E/I = 1$ highlights a student's extrovert psychological traits, while a value 0 indicates a more introvert personality. Such a same fashion applies to the remaining three dichotomies (i.e. Sensing versus Intuitive (S/N), Thinking versus Feeling (T/F), and Judgment versus Perception (J/P)).

Table 3.2 A sample of random data

Stu_id	Gender	E/I	S/N	T/F	J/P
S_1	1	1	0	0	1
S_2	0	0	1	1	1
S_3	0	1	1	0	0
…	…	…	…	…	…

3.3.1.2 Experimental setup

An exact method and a random method were selected to compare with the proposed discrete-PSO algorithm. The exact method is a brute force method that enumerates all possible solutions to the given group composition problem. It then evaluates each solution using the fitness function (cf. Equation (3-10)). In the end, the solution with the best fitness value is supposed to be the one we are looking for. Unlike other approximation algorithms, the final solution searched by an exact method is deterministic and can thereby act as a base line for others.

The random method is one of the traditional methods that are widely used to compose learning groups. Each group is composed by randomly picking students from the whole participation.

Different from the exact method, both the discrete-PSO algorithm and random method are a stochastic method, that is, every try of these methods could yield a different result. Thus, both methods ran ten times in an attempt to possibly cancel out biased results brought by a single try.

With regard to parameter settings, the group size was set to three. For the discrete-PSO algorithm, 200 iterations and 50 particles were chosen, $\omega = 0.1$ $c_1 = 0.2$ and $c_2 = 0.5$.

All three methods were implemented in MATLAB script language. The simulation experiment ran on a personal computer with Intel(R) Core(TM) i7-4600U CPU 2.10GHz and 8GB RAM.

3.3.1.3 Results

Computational performance: Table 3.3 shows that the discrete-PSO computed good formation (the fitness is almost equal to the optimal that the exact method always searches for). Although the time cost of the discrete-PSO grows as the number of students increases, the growth is still linear and much lower than the exact method. On a relatively slow computer, creating a group formation for 3000 students was possible in approximately 13 minutes.

The exact method's time cost grows exponentially as the number of students increases. This fact, however, is consistent with the analysis in Section 3.2.1, because it is essentially a NP-hard problem. The random method is the fastest one among the three by comparison. The fitness values it delivers are, however, much lower than the discrete-PSO.

Table 3.3 Computational performance of an exact method, a random method and the discrete-PSO algorithm

#students	**Exact Method**		**Random Method**		**Discrete-PSO**	
	Fitness value	Time cost (s)	Fitness value	Time cost (s)	Fitness value	Time cost (s)
9	0.7222	0.1005	0.6000	0.0006	0.7222	2.1955
12	0.6667	4.547	0.5347	0.0072	0.6667	2.5959
15	0.7778	3187	0.5772	0.0071	0.7728	3.0693
33	N/A	N/A	0.5795	0.0007	0.7467	5.7624
99	N/A	N/A	0.6014	0.0016	0.7561	15.9324
333	N/A	N/A	0.5802	0.0103	0.7164	54.9124
999	N/A	N/A	0.5820	0.0181	0.6760	193.2384
3000	N/A	N/A	0.5837	0.0407	0.6465	809.6901

A declining tendency of the discrete-PSO's fitness as the data size grows may imply us to narrow the margins against the random method when applying to much larger data sets. This is, however, a common issue of all approximation algorithms. Those algorithms merely explore a relatively smaller area of the whole problem scope compared to an exact method. Although solutions may reveal inferiority especially in large problem domains, they pay much lower time prize by comparison. Anyhow, their gains in a unit of time are still considerable. Moreover, such an algorithm can solve a larger problem that an exact method cannot cope with at all. In this case, to solve a problem of 15 students, the exact method needs almost one hour. It is very hard to imagine how much time it would cost for 3000 students. In principle, it is possible to maintain a considerable margin by tuning up the number of iterations and particles. It then certainly needs to pay more time prize in the end. All in all, a tradeoff between quality of solutions and time cost needs to be considered when applying such an algorithm.

Stability: the discrete-PSO algorithm is a stochastic method, which means that it is possible to see different results in different run times due to the varying initial particles and the random parameters for each run. To examine its stability (are the results of different runs in fact different or not?), a standard deviation of fitness values obtained in different runs was calculated. With a comparison of the discrete-PSO to the random method, the results indicate that the discrete-PSO algorithm is relatively stable (as shown in Figure 3.8, standard deviations between 0 and 0.01, compared to a range between 0.016 and 0.0826 for the random method), and that the number of students does not seem to have a major impact on stability. Based on this analysis, in a word, the discrete-PSO is a comparably stable method.

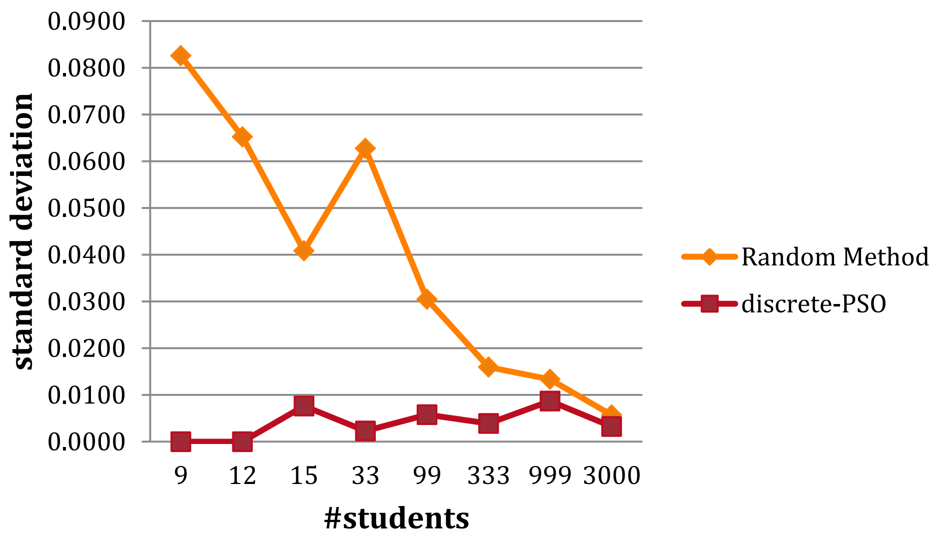


Figure 3.8 Stability of random method and discrete-PSO

3.3.2 Simulation experiment II: a comparison of four heuristics

In Section 3.3.1, the experimental results reveal the discrete-PSO's superiority to random methods in grouping quality and exact method in time cost. Still, some may ask a question of its computational performance over the other group composition algorithms that have been previously reviewed in this thesis, such as the genetic algorithm and ant colony optimization algorithm. To answer this question, another three heuristic algorithms were selected (i.e. a genetic algorithm, an adapted k-means clustering algorithm and an ant colony optimization algorithm) to compare their efficiency (i.e. grouping quality and time cost).

3.3.2.1 Data

The data was retrieved from iversity's two online surveys, namely motivation survey and demographic survey. I selected one question from each survey. One question reflects students' motivation to watch course videos (how many course videos do students intend to watch? cf. Appendix Ⅰ). The other collects students' geographic information (In what country do students currently live? cf. Appendix Ⅱ). I then selected a course with the most feedback to the selected two survey questions (1710 students). A sample of the data set is shown in Table 3.4.

Table 3.4 MOOC survey data set

User ID	Presumed video consumption	Living country
uid113XXX7	All	DZ – Algeria
uid113XXX3	Most	RU – Russia
uid120XXX6	Some	DZ – Algeria
…	…	…

Two grouping criteria were taken into account (one with heterogeneity and the other with homogeneity). First, students watching different amounts of course videos can be grouped together. The idea was to possibly encourage students to learn course knowledge from those frequent video consumers. If composing homogeneous groups of this attribute, there was a fear that those groups with many rarely-visited video consumers could not start effective course-related discussions because they are presumably lack of course-related knowledge. Second, students from close time zones were expected to stay with each other in the sense that they can share working time and thus possibly launch synchronous discussions. Note that night shift lovers would be exceptions.

The question in Appendix Ⅰ contains Likert-scale answers and Appendix Ⅱ presents country code to the answer list. Before feeding into the selected algorithms, numerical coding of those symbolic answers has to be done. For the likert-scale items, Section

3.1 already mentioned how to convert ordinal data into numerical data. For the geographic information, we converted the country code into time zone (in numerical number) using a third-party python library (i.e. pytz[13]).

3.3.2.2 Parameter settings

Group size is set to 4. Except for the adapted k-means clustering algorithm, the proposed discrete-PSO, the genetic algorithm and the ant colony optimization have a few parameters to set. Varying those parameter settings brings about different grouping quality. In order to tune such parameters so that those algorithms can perform their best, a grid search experiment was conducted upfront. Note that the ant colony optimization was completely borrowed from Graf's work (Graf & Bekele, 2006) and so were its parameter settings.

The mentioned grid search strategy works as follows. First, in order to narrow down the testing scope, retrieve each parameter's recommended values from the relevant literature. For example, for the genetic algorithm, the population size: $nPopu \in [30, 50]$, the crossover rate: $P_c \in [0.5, 0.9]$ and the mutation rate: $P_m \in [0.01, 0.1]$ (A. E. Eiben, Hinterding, & Michalewicz, 1999; Grefenstette, 1986; Safe et al., 2004). Second, give a grid of each parameter over the recommendation range. For example, $nPopu = \{30, 50\}$, $P_c = \{0.5, 0.7, 0.9\}$ and $P_m = \{0.01, 0.05, 0.1\}$. Third, enumerate all combinations of the given parameters' grids. For instance, $\{nPopu = 30,\ P_c = 0.5, P_m = 0.01\}$ is one out of those $2 \times 3 \times 3 = 18$ possibilities. Fourth, test each combination with the given MOOC data set and output the grouping quality. Finally, select a combination of parameters with the highest grouping quality.

Such a grid search experiment was conducted for both the genetic algorithm and the discrete-PSO. According to the experimental results (cf. Appendix Ⅲ and Ⅳ), $\{nPopu = 50,\ P_c = 0.9,\ P_m = 0.05\}$ for the genetic algorithm and $\{nPopu = 30, \omega = 0.1,\ c_1 = 0.1, c_2 = 0.5\}$ for the discrete-PSO were suggested, respectively.

3.3.2.3 Experimental setup

The selected four algorithms were implemented in Python and ran on an Ubuntu virtual machine (2 CPUs and 4098M memory, the host computer with Intel(R)

---

[13] https://pypi.python.org/pypi/pytz/

Core(TM) i7-4600U 2.10GHZ CPU and 8G memory). In case of any by-chance result, each algorithm ran 10 times. The grouping quality and time cost of each run time was retrieved.

All of the selected algorithms operate iterative optimization, that is, they need to run many iterations before achieving a near-optimal solution. In order to create a fair competition to all, a common stopping condition needs to apply. As the previous studies suggest (Qiang & Xiaoyan, 2007; Safe et al., 2004; Zielinski & Laur, 2007, 2008), two basic stopping criteria were applied in this thesis:

- An upper limit on the number of iterations;
- No chance of achiving a significant improvement in the next iterations.

The upper bound of the number of iterations was set to 100. To fulfill the second criterion, the program stops when the latest 2/3 iterations cannot improve the so-far-found best solution. However, solely relying on a combination of these two criteria often stops the program shortly after it starts. For example, if the first two iterations do not change the results, it then has to stop, which certainly terminates optimization too early to explore new solutions. Thus, a lower bound of the number of iterations (50 in this thesis) was also suggested to avoid running into such trouble.

3.3.2.4 Results

Fitness value and time cost: as shown in Table 3.5, ACO outperforms the others regarding the fitness value that indicates the average quality of the resulting groups (the higher the fitness value is, the better the average quality is). Of main interest is the proposed discrete-PSO algorithm. It only prevails over the genetic algorithm (0.5289 vs 0.5254, p-value < 0.001). The stability of all the selected algorithms tends to be close to each other and under a very small quantity (~0.001) when observing the standard deviation over ten run times.

Table 3.5 Fitness value

Runtime	Discrete-PSO	GA	Adapted k-means Algo.	ACO
#1	0.5268	0.5260	0.5427	0.5750
#2	0.5299	0.5241	0.5403	0.5755
#3	0.5283	0.5250	0.5415	0.5754
#4	0.5290	0.5265	0.5423	0.5747
#5	0.5256	0.5249	0.5411	0.5758
#6	0.5309	0.5260	0.5426	0.5765
#7	0.5280	0.5278	0.5415	0.5750
#8	0.5287	0.5253	0.5383	0.5760
#9	0.5328	0.5242	0.5401	0.5754
#10	0.5299	0.5242	0.5387	0.5762
Ave.	0.5289	0.5254	0.5409	0.5755
SD	0.0019	0.0011	0.0014	0.0005

When taking time cost into account, the GA might be the best choice because it costs comparably less time (cf. Table 3.6). ACO, however, took much more time to compute the results than the others. If looking at time cost and grouping quality as a whole, we can draw a conclusion that higher quality pays more time price in the end. None of the selected algorithms can escape from the tradeoff between quality and time cost.

Table 3.6 Time cost (in seconds)

Runtime	Discrete-PSO	GA	Adapted k-means Algo.	ACO
#1	944	325	3576	8390
#2	958	330	3579	8427
#3	922	355	4351	8130
#4	948	360	4376	8355
#5	987	325	4048	7882
#6	1009	329	4057	8068
#7	811	342	3890	7924
#8	835	350	3896	7967
#9	996	337	3612	8918
#10	1003	349	3618	8959
Ave.	941	340	3900	8302
SD	65.1	12.2	291.0	367

Difference between groups: apart from fitness value and time cost, another interesting indicator is the difference among the resulting groups. The goal of grouping is not merely to optimally satisfy grouping criteria, but also to eliminate the average difference among the resulting groups as much as it can. As defined in Equation (3-3) and (3-4), GCP and GFQ stand for a group's quality and a group formation's quality respectively. We can then calculate the average difference among the resulting groups as Equation (3-17), assuming M groups in total.

$$\boldsymbol{GAD = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(GCP_i - GFQ)^2}} \qquad (3\text{-}17)$$

As we can see in Table 3.7, this indicator does not tell a big difference (around 0.08, no more than 0.1) and the standard deviation over ten run times is much small (no more than 0.002). We can conclude that they perform close to each other regarding average difference between the resulting groups.

Table 3.7 Average difference between the resulting groups

Runtime	Discrete-PSO	GA	Adapted k-means Algo.	ACO
#1	0.0788	0.0798	0.0902	0.0801
#2	0.0809	0.0798	0.0939	0.0795
#3	0.0797	0.0808	0.0956	0.0791
#4	0.0829	0.0799	0.0972	0.0803
#5	0.0841	0.0818	0.0953	0.0795
#6	0.0806	0.0772	0.0977	0.0803
#7	0.0810	0.0829	0.0934	0.0796
#8	0.0779	0.0803	0.0953	0.0825
#9	0.0803	0.0766	0.0966	0.0820
#10	0.0831	0.0804	0.0950	0.0802
Ave.	0.0809	0.0799	0.0950	0.0803
SD	0.0018	0.0017	0.0020	0.0010

Heterogeneity and homogeneity: recall that, in this simulation experiment, we chose to group students with heterogeneous video consumption and homogeneous time zone. So far, we looked at the fitness value that is an overall measure of the heterogeneity and homogeneity. Yet, how heterogeneous and homogeneous the examined attributes are in the resulting groups has not been investigated.

By definition, when a value goes down closer to 0, its homogeneity reveals more. In contrast, as it approaches to 1, its heterogeneity grows accordingly. As shown in Figure 3.9, first, ACO holds the best homogeneity inside the groups, yet the worst heterogeneity. Second, the adapted k-means algorithm outperforms the others in heterogeneity. These two points in conjunction with their fitness values in Table 3.5 provide us insight that ACO and the adapted k-means algorithm's superiority to the other two relies on their capability of diminishing homogeneity and increasing heterogeneity inside groups respectively. Third, the discrete-PSO wins the genetic algorithm by a slight margin in either homogeneity or heterogeneity. Besides, another evident point is that the heterogeneity that all algorithms made seems too small (around 0.2) to reveal any heterogeneity.

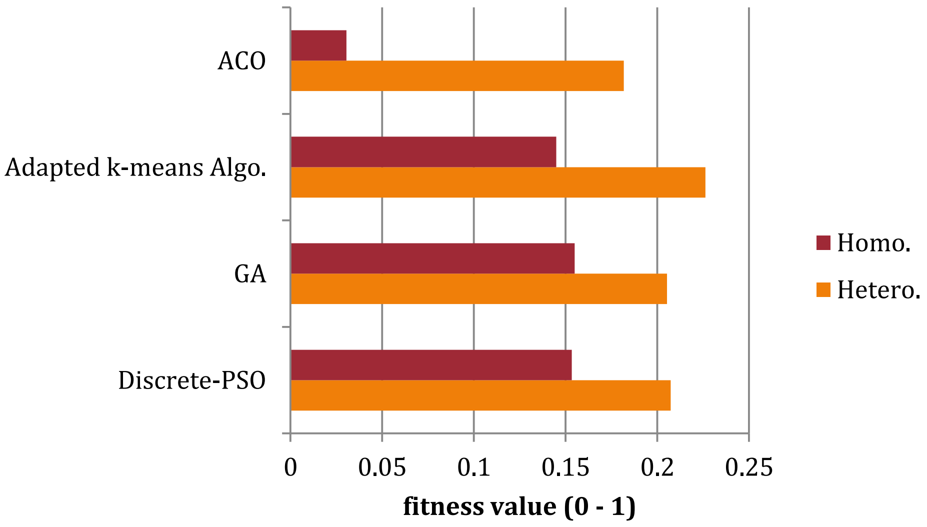


Figure 3.9 Heterogeneity and homogeneity in groups. Here is the average heterogeneity and homogeneity over ten run times (cf. Appendix Ⅵ)

To investigate reasons why the heterogeneity in the groups is not noticeable, we need to look into the data source. As shown in Figure 3.10, the answers to video consumption (cf. Appendix Ⅰ) are visualized. 63.5% students intended to watch all videos. Roughly one quarter of students planned to watch most. Very few students, however, attempted to skip most videos. This skewed data in the end resulted in that many students of almost same characteristics go to a group (either with most or all video consumption) because there are not many other options. Students of different characteristics are rather scarce. As such, heterogeneity (measured with peer distance) in each group became understandably small.

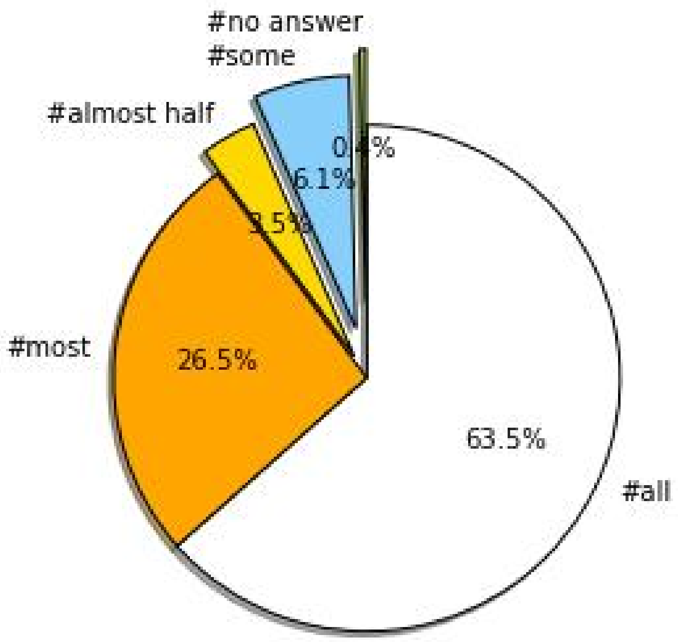


Figure 3.10 Distribution of the answers to video consumption. The percentage stands for how many students answered 'all', 'most', 'almost half', 'some' or gave no answer to the survey question in Appendix Ⅰ

3.3.3 Discussion

Through two simulation experiments mentioned in Section 3.3.1 and Section 3.3.2, we have an impression of exact methods, random methods and four different heuristics. Still, what is the suggestion to select those algorithms? To answer this question, first of all, we should realize the tradeoff between grouping quality and time cost. The exact method guaranties grouping quality, but the time cost grows exponentially as the class size increases. This method should certainly be out of our consideration when composing small groups in current MOOCs. Random methods run quite fast because they do not look at grouping criteria and thus the grouping quality is understandably the lowest. Can we make grouping quality better? Yes, the selected heuristics were proposed to fulfill this goal. But, again, they cannot run beyond the tradeoff either. Selecting one out of those four heuristics should refer to at least two points. First, time cost should be affordable. The ant colony optimization algorithm takes much more time than the other three, though it makes the best grouping quality in the experiment. With only 1710 students and two grouping attributes, it took more than two hours. Certainly, 1710 students are far below the size of the most MOOC courses. When facing far more students, it would cost much more time. This, in the end, would affect its ability to be scaled up. Comparably, the adapted k-means clustering algorithm could be a good choice due to its relatively lower time cost and slightly worse grouping quality. Second, we should also consider grouping criteria. As argued in Section 3.2.2.5, the ant colony optimization algorithm and adapted k-means clustering algorithm can work for the heterogeneous and homogeneous grouping criteria. However, for some criteria of constraint type or even more complex ones with a structure of many constraints, both of them might not easily solve the problem. In such cases, the discrete-PSO algorithm and genetic algorithm can be taken into consideration because of their immunity to those types of grouping criteria. With regard to grouping quality, the former is superior to the latter.

The simulation results may imply some possibilities to improve the ant colony optimization algorithm. As we already saw, the biggest problem of the algorithm is its high time cost. The high time cost partly derives from the expensive 2-opt local search which embodied in the algorithm. If we can apply the same local search as the adapted k-means clustering algorithm rather than the 2-opt local search to the ant colony optimization algorithm, the time cost could be considerably reduced. After all, as we can see, the adapted k-means clustering algorithm cost much less in time.

3.4 Applying a grouping algorithm to a MOOC

Up to now, we have already studied computer algorithms to compose learning groups. Through two simulation experiments, the upsides and downsides of each selected algorithm have, thus far, been unveiled. Except for a lab environment, it would be interesting to test those grouping algorithms in a real MOOC environment too. For this reason, in the year of 2014, I conducted a study on our industry partner's MOOC platform, iversity.org. Note that the original work was published in (Zheng, Vogelsang, & Pinkwart, 2015). In that study, the adapted k-means clustering algorithm was used to compose small learning groups in a MOOC. The study sought to explore how to practically compose small learning groups in MOOCs and its impact on drop-out and learning performance (see the research question RQ2 in Section 1.2).

3.4.1 Methods

3.4.1.1 Course description

We chose the second iteration of the course “The Fascination of Crystals and Symmetry”, which was offered on the iversity.org platform. It is an introductory course to crystallography held by Dr. Frank Hoffmann (University of Hamburg) and ran twice on the platform: In a first iteration from April to June 2014 and in a second iteration from October to December 2014.

The course systematically examines and discovers the concept of symmetry using the example of a crystal’s atomic structure. A total of 12,661 students registered for the first iteration, 1,326 of whom actively engaged with the course material (meaning they achieved more than 5% course progress by watching lectures or performing assessments).

After having observed the successful first iteration of the course, we chose the second iteration for our grouping experiment. An important factor for this decision was that the course offered quizzes, three peer graded homework assessments, open discussion questions within the course material and a paid certificate track together with an instructor graded final exam. In particular, the open discussion questions seemed well suited to engage students in group interaction.

The second iteration was smaller (3,209 enrolments) but had a higher percentage of actively engaged students (771, i.e. 24.03%). All mentioned numbers reflect the enrolment and progress status as of January 2015.

3.4.1.2 Experiment procedure

In this study, we came up with a grouping scenario that is set up quickly by making use of existing tools, and is thus very easy to reproduce in similar online learning contexts. The main collaboration tools used in this study are available for free (Facebook, Email, Skype and Google+). According to students’ preference, local groups are also optional.

After the start of the course, 80 percent of the participants received an email with a short survey asking for information about their demographics, personality, course access and learning group preferences (cf. Appendix Ⅴ). The remaining 20% of the course participants did not receive this survey and were not grouped (namely, NoG-CG), but served as a control condition. Students who provided sufficient answers to the survey (namely, AlgoCG) were put into different types of groups with the fixed size of 10 and received a second email a few days later. This second email asked them to join their group and presented the other group members with the personal description given in the survey (cf. Appendix Ⅴ, Question 11). The email also contained a link to the first open discussion question in the course material and a link to their group (if applicable). Some of those survey receivers who did not answer were put into a NoG-NA condition and those with insufficient answers were assigned into a NoG-IA condition. Besides, some of those survey receivers who did not answer but presented their facebook accounts were assigned into random groups with group size of 10 also (namely, RandCG). The student conditions might look complex. We only need an algorithmic grouping condition, a random grouping condition and a control condition. But, in real MOOC environments, some factors could affect such classification. For example, in this experiment, some students chose to answer the questionnaires, some gave insufficient answers and the others decided not to do so. We can simple ignore students' different motivation to the questionnaires. But we found the responsiveness to the survey also interesting to be watched (see analysis in Section 3.4.2.1), and we therefore decided to separate those conditions.

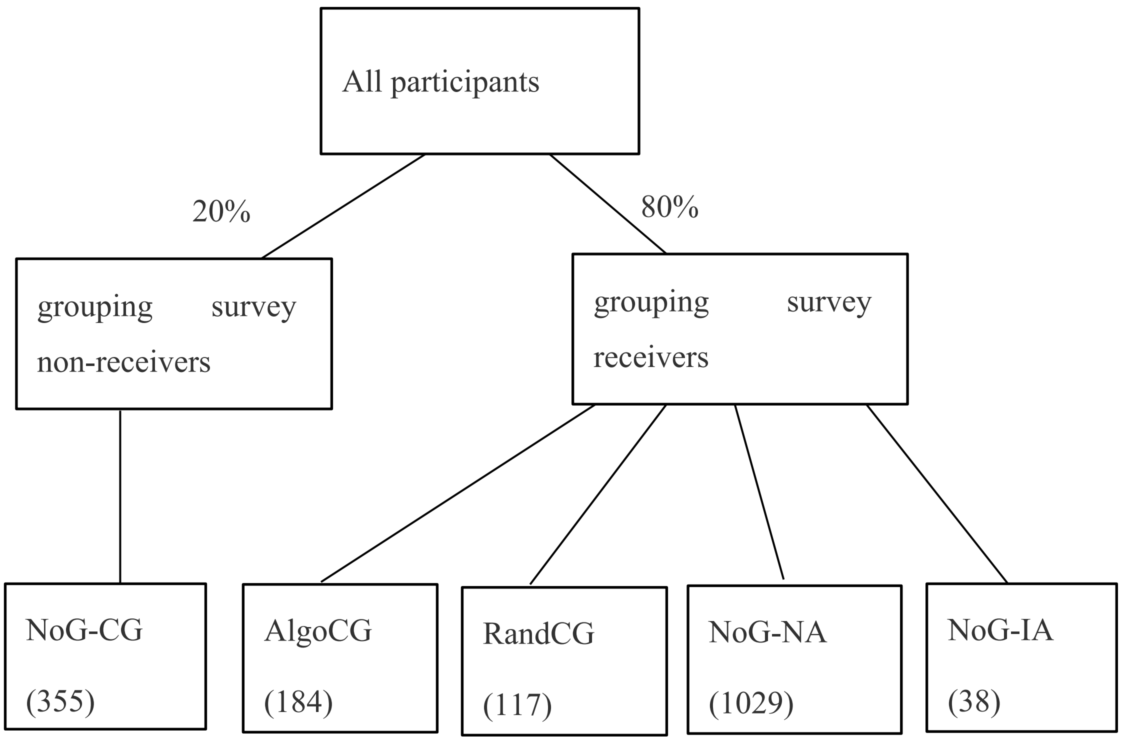


Figure 3.11 Students' conditions

As a last grouping related intervention, we sent a small post grouping survey by the end of the course. This survey was only sent to the 80% who had also received the initial grouping survey and contained questions about satisfaction with and intensity of the group work.

3.4.1.3 Grouping data

The data set was originated from the students' answers to the grouping survey we sent out shortly after the beginning of the course. The grouping survey contained 11 questions as shown in Appendix Ⅴ. We designed this survey in order to extract information about students' gender (question 8), time zone (questions 3 and 4), personality (question 10), learning goal (question 9) and language (questions 6 and 7) and fed this into our grouping algorithm. A snapshot of the data set is shown in Table 3.8.

Table 3.8 Grouping data set (derived from the grouping survey)

Stu_id	**Gender**	**Time Zone**	**Personality**	**Learning goal**	**Language**
1	Male	Berlin	Answer 1	In general	English, German
2	Female	Lucknow	Answer 2	In-depth	English
3	Female	Portland	Answer 3	In-depth	English
…	…	…	…	…	…

Besides, students' preference to online or local offline groups (question 5), their preference about collaboration medium (questions 1 and 2) as well as an introductory text (question 11) were collected.

3.4.1.4 Grouping algorithm

Based on the collected responses from the grouping survey, we first segmented the respondents into five classes according to their collaboration preferences (questions 1, 2 and 5), namely local, Facebook, Google+, Skype or Email group. The option meetup.com was underrepresented in the responses and thus excluded. For each class, we then extracted each participant's demographical, psychological and geographical information (i.e. gender, time zone, personality, learning goal and language). Next, a group composition algorithm was applied to compose learning groups of 10 students in each taking into account heterogeneities (namely gender, personality and learning goal) and homogeneities (i.e. time zone and language). Since the adapted k-means clustering algorithm (cf. Section 3.3.2.4) performs better in grouping quality than the genetic algorithm and the discrete-PSO algorithm, it was selected to compose those algorithmic groups. It, on the other hand, costs more time. However, in this experiment, very few grouping subjects (184 students in total) made it affordable. Note that local groups were meant to only contain students from the same cities, resulting in very few and small groups qualifying for this option.

3.4.2 Experimental results

3.4.2.1 Drop-out rate and survey responsiveness

We assigned the student groups approximately one week after the official start date of the course. Here, the definition of a 'drop-out' is any student who did not submit any quiz or assessment and thereby did not qualify for any course score – after this group assignment. The drop-out rate is then the percentage of students who drop out.

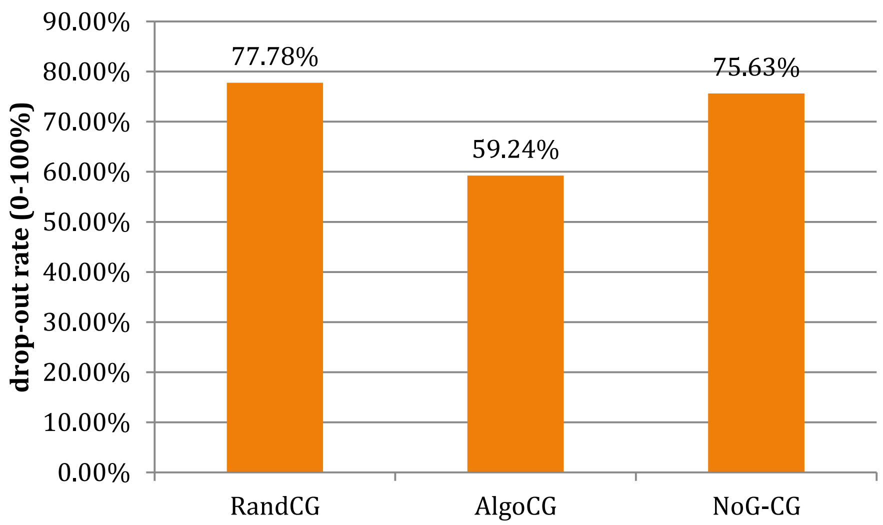


Figure 3.12 Drop-out rate

As shown in Figure 3.12, a first glance at the results reveals that, AlgoCG, the condition of our main interest, appears to be the best performing condition regarding drop-out rate, with a considerably lower rate (59.24%) compared to random grouping (RandCG, 77.78%) and the controlled no-grouping condition (NoG-CG, 75.63%). The impact on drop-out is thus evident. The algorithmic grouping shows its capability of reducing the drop-out rate in comparison with a random grouping strategy and a typical MOOC circumstance without any grouping. But the random grouping does not reveal its benefit over the no-grouping condition. This might suggest to us the necessity of taking students' information into account when composing online learning groups.

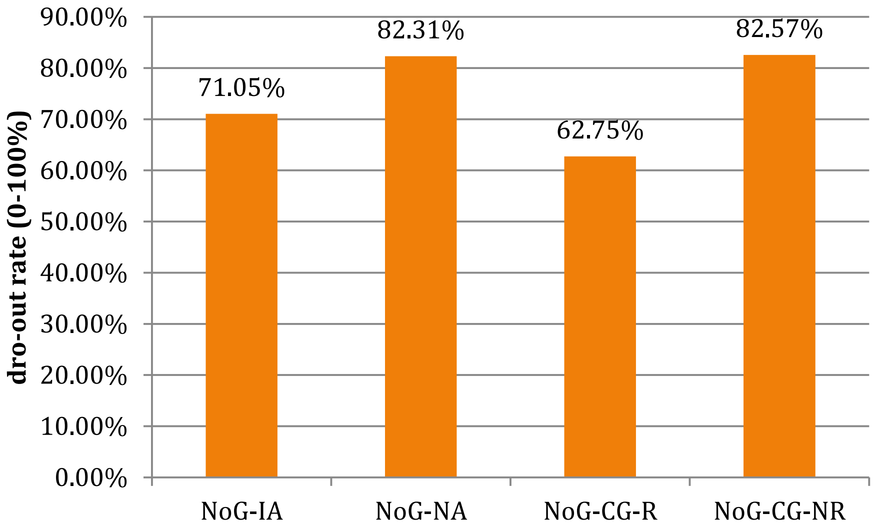


Figure 3.13 Survey responsiveness' impact on drop-out rate. NoG-CG-R represents those students who stayed in the control condition but responded to the motivational survey whereas NoG-CG-NR stands for their counterpart who did not

Nevertheless, another hidden factor is of interest to be unveiled as well. It might be known to many that students in the algorithmic grouping condition had signaled their willingness for learning groups by filling out the grouping survey whereas those in the random grouping condition had rejected it. This fact might imply that students in the former condition had a higher level of engagement in groups than those in the latter condition. Therefore, there is no surprise that proportionally fewer students drop out from the algorithmic grouping condition. When analyzing the interplay between grouping condition and drop-out rate, survey responsiveness might thus need to be watched very closely also. This attracted our attention to take a second glance at the results. As shown in Figure 3.13, first of all, those who gave insufficient survey responses seem to be less likely to drop out than those who did not respond at all, (NoG-IA: 71.05% vs NoG-NA: 82.31%). Second of all, in the control condition without grouping, those who interacted with a motivational survey (regularly sent to users by the learning platform) had a considerably lower drop-out rate than those who did not (NoG-CG-R: 62.75% vs NoG-CG-NR: 82.57%). We can conclude that non-responsive students (with regard to a survey) are more likely to drop out than responsive students, which, meanwhile, implies that survey responsiveness, as an uncontrolled factor, could intervene in the impact of grouping on dropout.

3.4.2.2 Learning performance

In order to analyze the experiment's impact on student's learning performance, we looked at students' scores on quizzes and homework. Figure 3.14 visualizes average as well as minimum and maximum scores within the various experimental conditions. If we look at the average scores, the algorithmically composed groups do not significantly outperform those from other conditions. We thus do not find evidence for a positive impact of any condition on learning performance as measured by score. A one-way ANOVA implied no statistically significant difference between the conditions. This, in the end, tells us that the findings so far do not indicate any of grouping's positive impact on learning performance.

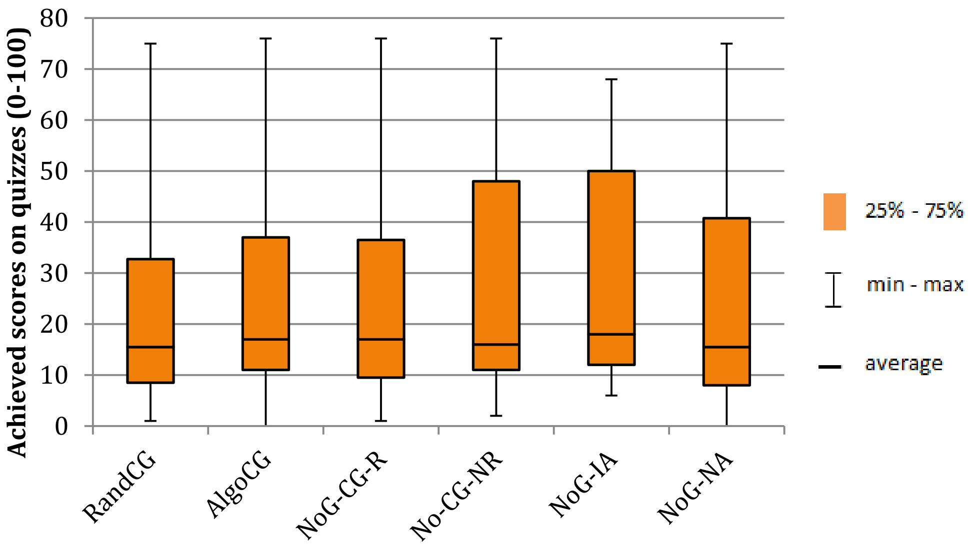


Figure 3.14 Box plot of learning performance

3.4.3 Discussions

The results of this MOOC experiment are biased and the noise could come from two sources. First, self-selection, as a hidden factor, could play an important part in yielding such results. While only those who successfully completed our grouping survey were assigned to the AlgoCG condition, we chose to compose RandCG from students who did not respond to that survey. Thus, one can interpret the results in a different perspective: fewer students drop out from the algorithmic groups than the random groups because students in the algorithmic groups apparently have more motivation to small learning groups than their counterparts in the random groups. To avoid such an effect, we should have chosen some students from those who answered the survey and composed them into random groups. Since the number of survey respondents was too small (merely 184 students answering all survey questions), we did not split this small amount of students into two conditions. The second source is the by-chance instance. Since this experiment was only conducted once in a MOOC course, even though the results indicate lower drop-out rate in the algorithmic groups, we still cannot say the difference is statistically significant. To mitigate this problem, more and more evidence has to be collected from different MOOC courses and finally a statistical analysis needs to be conducted based on those evidence. The positive side is that this experiment is easy to be reproduced in other MOOCs. Thus, collecting such evidence is not a problem of method but rather of time.

Another questionable aspect of the experimental results is the fact that no actual collaboration was observed in the Facebook and Google+ groups. How can small learning groups have an effect if nothing is going on in the groups? Some students claimed to have collaborated in the post grouping survey, but we lack information about what happened in the local, email and skype meetings. As such, we do not have a proof that any group work actually happened. This, in the end, inspires two lines of thinking. First, why do students rarely use such social media as Facebook and Google+ to launch course-related discussions? Using Facebook to support learning is not new (C.-C. Hsu et al., 2014; C.-C. Hsu, Chen, Huang, & Huang, 2012; C. C. Hsu, Chen, Chang, & Huang, 2012; Kirschner, 2015; Manca & Ranieri, 2013). We may expect those social media can solve the problem of students' communication. But, as Kirschner pointed out (Kirschner, 2015), it might not be effective for knowledge construction. It likely implies that a learning tool with a special design for collaborative learning could be needed. Another line of thinking is how to avoid students interacting with each other beyond the expected tool. The fact might not suggest to us a need to completely avoid such a case. In this day and age, many communication tools can be leveraged to contact another people or a group of people. A single student may have a few tools on different devices and he/she would prefer to use which tool according to convenience. For example, he/she could join a skype meeting at home and perhaps text their group members via another messaging tool in the library. Clearly, this would challenge data analysts if students contact others via multiple channels. Because it is difficult to collect all the data from different tools and fuse those data.

3.5 Chapter summary

This chapter elaborates the approaches to solve the group composition problem (cf. Section 3.2 and 3.3). Because the group composition problem is a NP-hard problem, an exact method is too expensive to seek an optimal solution. As the problem size grows, the time cost increases exponentially. As such, exact methods are not suggested to compose learning groups in MOOCs due to their large enrolments. Four heuristic algorithms can be used to mitigate the high time cost problem of exact methods. When testing on a real MOOC data set, their computational performance was unveiled in a comparable fashion. The ant colony optimization algorithm performs the best in grouping quality. But it took much more time than the others (it took more than two hours for only 1710 students). When the problem scales up, the high time cost is still a worry. As discussed in Section 3.3.3, another important factor to selecting a grouping algorithm is grouping criteria. If the grouping criteria are beyond the scope of the heterogeneous and homogeneous type, the discrete-PSO algorithm would be a good choice. These findings give an answer to the research question RQ1.

In addition, this chapter describes a grouping experiment in a real MOOC course (cf. Section 3.4). This experiment does not need any support from additional learning tools and is thus easy to be reproduced in other MOOCs. It leverages social media (e.g. Facebook and Google+) as the group interaction medium. Through observing the course participants' dropout and learning performance, the algorithmically composed groups reveal their capability of reducing the drop-out rate in comparison with a random grouping condition and a control condition (no grouping). However, the learning performance among those three conditions is not significantly different. This, in the end, answers the research question RQ2.

4 GROUP RE-COMPOSITION: FROM APPROACH TO IMPLEMENTATION

This chapter starts to elaborate on another important concept of this thesis: *Group Re-composition*. Addressing the problem of group composition, Chapter 3 already presented four group composition algorithms and discussed their strengths and weaknesses. Apparently, such group composition algorithms bring teachers many benefits. Not only can they help teachers fulfil their grouping goals (e.g. ability mixed learning groups), but also, in recent MOOCs, using those algorithms to compose learning groups might be able to reduce student attrition according to the aforementioned MOOC experiment (cf. Section 3.4). Nevertheless, any attempt to blend learning groups into current MOOC-related didactics might still confront many practical difficulties, for instance, students' reluctance to respond to grouping surveys that are usually used as an important data source for group composition. No different than any of those online surveys occasionally yet intentionally flying into our email boxes, online students do not have any obligation to answer them either, not to mention that some of those survey questions might expose personal privacy (e.g. gender and age). The feedback rate is thus understandably low in many cases. Yet, this would, in the end, strike a fatal blow to the group composition algorithms. After all, no algorithm has thus far been capable of composing learning groups in a cold-start mode (requiring nothing of students' information at all). Another, yet more serious, problem is group instability. Group instability could be caused by the widely reported high drop-out rate, that is, many students drop out, which results in many groups incomplete in size. If they still need to pursue their group work, the left-behind human resource would be far too short to achieve the intended group goals. It would be better to take one more step beyond group composition if considering a more sophisticated way to create and maintain learning groups in up-to-date MOOC-like large online learning environments. This chapter intends to introduce a group re-composition approach on top of the current group grouping methods in hope of overcoming the practical difficulties that recent large online open learning environments could bring about. Meanwhile, a second hope is to shed some light on the areas that the current group composition methods have not covered so far.

This chapter mainly approaches a group re-composition method, its effect in principle and its implementation in MOOCs. Section 4.1 describes the group re-composition approach in detail, its differences from the current group composition methods and what problems it seeks to solve. Next, Section 4.2 and Section 4.3 elaborate on a group tool (namely, twoleaves) that fulfils the goal of putting the proposed approach into practice. Finally, Section 4.4 summarizes the whole chapter.

4.1 A dynamic group re-composition approach

As mentioned at the beginning of this chapter, students' reluctance to fill out a grouping survey could, to some extent, diminish data source and thus pose difficulties to group composition in open online learning environments. Even though this can be solved by combining random grouping strategies (because those do not rely on data presence), maintaining complete learning groups is still a great challenge because the high drop-out rate would stop many students from their current groups. Thereby, this thesis tries to argue that dynamic group re-composition would then be an option to overcome these difficulties. The following starts to look into the mechanism of group re-composition.

4.1.1 Schema

What is Dynamic Group Re-composition? Dynamic Group Re-composition, literally, has twofold aspects. The first regards group re-composition. This does not differ highly from similar operations in sport teams. Typically, basketball teams (e.g. NBA) often rebuild their teams after a playing season via buying new members in and selling some members out if necessary. Aside from the business considerations, another important motivation behind such operations is believed to promote group performance. Teams are very complex social entities. Many factors could cause a setback on the way to success, for example, individuals' under/over-expected skills and peer conflicts. Likewise, learning groups cannot escape from such either. Rather, the situation could be even more serious because many students are very new to the target online courses. We cannot see their performance in groups until they start group work. Yet, once you see it, you may find it too late to change the group composition. On the contrary, in NBA teams, many of those players would have already revealed their performance to the public. Most importantly, re-composing groups is allowable. Inspired by those sport teams, such a group re-composition operation could also be applied to learning groups. As group processes, we can gain insights regarding group cooperation, group outcomes and individuals' development and then apply such knowledge to rebuild groups in hopes of bringing about better composition of groups.

The second aspect lays stress on a dynamic process to recompose groups. This dynamic process, in fact, addresses the question of how often group re-composition should take place. A dynamic process could easily be misunderstood as changing group formation every second. However this could potentially hurt group learning. Learning groups, as very complex social entities, basically need much time to process so as to achieve the group effectiveness. As in (Tuckman, 1965; Tuckman & Jensen, 1977), Tuckman et al. pointed out the five stages (i.e. forming, storming, norming, performing and adjourning) that many small groups go through. If one changes group formation during the sequence of group development, that could probably turn out to jeopardize the widely recognized group processes. As such, this thesis chooses to situate the concept of group re-composition in a multi-task online course setting, that is, we recompose groups when a new task comes in the selected courses. Each group task can be considered as an independent unit, which means each task needs to have its own goals. Group re-composition can be conducted during the transition of each pair of consequent tasks and thus does not hurt the wholeness of group development in each task. In process of any ongoing task, students are not allowed to leave for another group.

Concretely, as depicted in Figure 4.1, we can start the first group task with an initial group formation. These initial groups can be composed via the employment of the group composition algorithms mentioned in Chapter 3 if participants' data (e.g.

demographic data or motivation data) is available. Otherwise, leveraging a random method could be the only choice. In principle, those random groups could be inferior to the groups composed by a group composition algorithm. However, random grouping could be one of the most realistic approaches when facing many online students who are cautious to expose their data. In practice, a combination of algorithmic methods and random methods could be an optimal option, that is, we can do algorithmic grouping for data contributors (i.e. students with data) and random grouping for students without information. After all, data scarcity or even unavailability in real cases would not threaten the approach anymore because this initial group formation is not unchangeable. Rather, it can be improved in the following task(s).

Next, we retrieve group data from the first group task. This data mainly comes from two sources: group interactions and group success (e.g. learning performance). Based on group interaction data, we can further detect individual student's characteristics (e.g. roles) in his/her group using structural analysis (e.g. Social Network Analysis). Together with group success data, we can then apply machine learning methods to induce group composition rules that indicate which characteristics (combined together) make successful groups or weak groups. Those generated composition rules are employed to suggest new groups for the next task if necessary. Through this iterated process, we can learn group composition knowledge from the data and such knowledge, in principle, updates and accumulates task by task. For each new task, the up-to-date group composition knowledge is applied to recompose the learning groups.

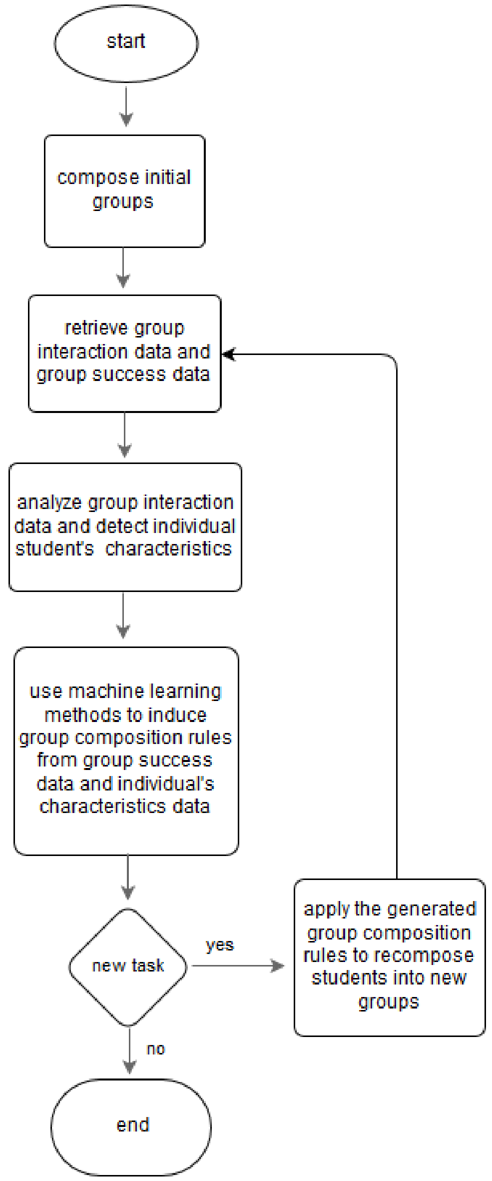


Figure 4.1 Schema of the proposed group re-composition approach

4.1.2 Ties to group composition

A close look at the whole dynamic group re-composition process might reveal the fact that group composition actually occurs multiple times. The whole process can be simplified as shown in Figure 4.2. It starts with group creation and then updates relevant data (e.g. students' characteristics and group outcomes). If a new task follows up, we create new groups with the up-to-date data. If not, we terminate the whole process. Clearly, the group re-composition mechanism runs on top of group composition. It needs to make use of the module of group composition to make groups anyhow. However, their distinctions appear to be evident too. First, students' data is no longer too static to reflect students' personal development along group process. For example, students' ability, which is widely used in ability grouping, could change over time, either towards better levels or worse (Moreno et al., 2012; Robert E. Slavin, 1987). Certainly, such data as gender and age would be exceptions. Second, when recomposing groups, it does not simply reapply the selected grouping criteria (e.g. heterogeneous roles groups) one more time. Rather, it induces and updates group composition rules task by task and applies them to the upcoming new tasks.

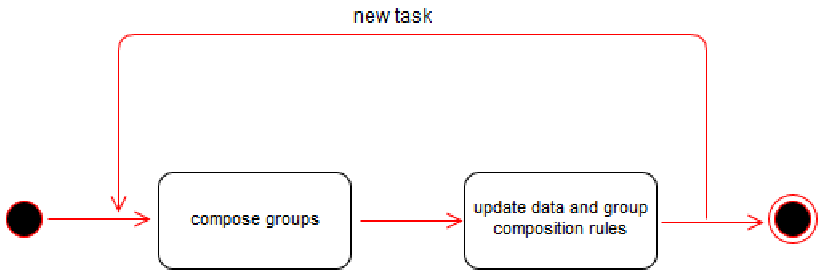


Figure 4.2 a simplified work flow

4.1.3 Problems to solve

With regard to problems addressed by this proposed group re-composition approach, at least the following points can be covered from a theoretical view. First, the usage of group re-composition could overcome the difficulties that the high drop-out rate brings to online learning groups. The high drop-out rate can result in the groups' incomplete size and thus the left-behind human resource in some of those involved groups would probably not be enough to pursue the new tasks. If the group re-composition approach is employed, those involved students can find a chance to leave for new groups rather than hopelessly stay alone.

Second, as a side effect, group conflicts (often caused by peer dissatisfaction) could also be resolved. Certainly, this is a passive solution rather than an active one. It can simply avoid those conflicted peers composing into the same group. In classrooms, teachers often play an important role in investigating the root causes of group conflicts and give some solutions accordingly. In large-scale online learning environments, little of such interventions can factually be made by human teachers when facing massive students. Thereby, this passive strategy could be a more realistic option in some cases.

Third, using this approach, students' very low response rate to (approx. 13% as shown in (Zheng et al., 2015)) the data collection procedure that is normally conducted before group creation could be less threatening. So far, most existing group composition methods rely on such pre-collected students' data. In an experimental environment, this would not pose any problem because you can choose to compose groups only for those effective responders. However, imagine that you, as a teacher, prepare to carry out learning groups for the whole course enrollment. This small amount of data would imply that you have to blindly compose learning groups for the vast majority and thus it would turn out to be no different than a random grouping method. This group re-composition strategy does not merely make use of the static data that gathered before the initial group creation, but also leverages interaction data during group process. If many students would not like to contribute group interaction data either (they still choose to keep silent), this approach would not be immune to the data scarcity problem either. But one more data source is still better than relying only on the static data.

Fourth, it does not rely on the existing grouping criteria (that are often too generalized and scenario-prone). Rather, this approach is totally data-driven and directly reflects the truth encoded in the data. It seeks to discover group criteria and reapply them over time instead of applying the pre-defined grouping criteria repeatedly.

Fifth, it accounts for group dynamics. For example, if somebody's role changes during group process, the data would accordingly reflect such. It is easy to be extended to other individual characteristic as well (e.g. students' ability).

Last but not least, the resulting group composition criteria that are captured dynamically along the course's whole lifetime could be very useful. For instance, it could be applied to suggest group assignment in some other similar learning contexts.

4.1.4 Feasibilities

The proposed group re-composition approach is a data-driven method. Data volume is thus vital to the success of this approach. This is the very reason for choosing current MOOCs as an application scenario. A MOOC course can typically enroll thousands of students, which, in principle, lessens the worry of small data volume. On the contrary, choosing traditional classrooms or on-campus moodle courses is relatively risky because of their much smaller class size.

Still, putting the proposed method into practice requires a lot. First, not every current MOOC course could be considered as a good fit. Rather, the method is apt for those with multiple team tasks. Second, it relies on many recent computer-supported technologies. Centering on these two aspects (namely, the course-related aspect and the technology-related aspect), the following sub-sections start to discuss its feasibilities.

4.1.4.1 Course-related analysis

What courses? When applying the method in a real course, team-based courses would be the scope. Nowadays, many educational institutions have actually realized the importance of teamwork when teaching students, for instance, in business and computer science schools (Kliegl & Weaver, 2013; Lingard, 2010). If we go to MOOC LIST [14] (a web application that list MOOCs offered by the best universities and entities) and search key terms “business” and “software development”, we can see plenty of course entries in the resulting list. Those courses can be considered as the target courses. Note that there should be much more courses beyond business and software development in favour of teamwork.

What tasks? First of all, the method runs an iterative mode as shown in Figure 4.1, which needs more than one group task. The hope is to learn group composition knowledge from the current task and then apply the obtained information to the upcoming task (s). Second, since group interaction data is vitally important to infer students’ individual characteristics, the assigned tasks need to be open-ended enough to encourage active discussions. In contrast, those tasks welcoming determined answers might be more likely to kill group discussions as long as the answer is given by someone.

Course owners or teachers normally decide their course content. Which tasks should be assigned in the course is factually a concern of their end. In order to make the best use of this group re-composition approach, keeping a partnership with those teachers becomes very important to its success as well.

4.1.4.2 Technology-related analysis

The proposed method requires such technologies as Social Network Analysis (SNA), machine learning methods and grouping algorithms (i.e. group composition algorithms). These technologies, in recent years, have increasingly been known to many scholars and professionals.

SNA in this method is used to detect students’ roles by looking into their interaction data. Roles identification has been studied for years. Studies have, thus far, suggested at least three approaches, namely, cluster analysis, machine learning methods and

[14] https://www.mooc-list.com/

social network analysis. In (Aviv, Erlich, Ravid, & Geva, 2003), Aviv et al. addressed such a problem using cluster analysis, which firstly groups students into several clusters, and then maps each cluster into a role according to its typical patterns. The mapping, however, had to be done manually back then. Chang et al. applied machine learning methods embedded in the IBM text miner to detect students' functional roles (Chang, Chen, & Wang, 2011). Likewise, Erlin et al. proposed a SVM method to identify students' roles (Erlin, Rahmiati, & Rio, 2014). Both studies apply machine learning methods and analyze the text content. Nevertheless, their approaches rely on human's manual intervention, more or less. Thanks to the development of SNA (Wasserman, 1994), the roles identification problem has recently been addressed by structural analysis via SNA (Marcos-García, Martínez-Monés, & Dimitriadis, 2015; Sundararajan, 2010). SNA does not merely show its distinctive advantage of structural manifestation of social relations among participants, but also advances the previous semi-automatic roles detection into an automatic stage.

A critical question to be answered when one applies SNA to detect group roles is how to map Social Network (SN)'s metrics into specific roles. A wealth of previous studies have begun to address such. Marcos et al. successfully identified isolated students, student-coordinators and teacher dependent students by observing SN's degrees, closeness and betweenness centrality (Marcos-Garcıa, Martınez-Monés, Dimitriadis, & Rodrıguez-Triana, 2008). They recently extended it to detect more roles (e.g. teacher-facilitator) via applying their SNA tool, Role-AdaptIA (Marcos-García et al., 2015). Rabbany et al. visualized the leaders and peripheral students using their Meerkat-ED tool (Rabbany, Takaffoli, & Zaïane, 2011). Brokers as an important role in social networks were also studied by Stuetzer et al. using SNA, and its SN characteristics have so far been uncovered (Stuetzer, Koehler, Carley, & Thiem, 2013). More recently, Capuano et al. revealed elaboration of mapping between roles and SNA metrics (Capuano, Mangione, Mazzoni, Miranda, & Orciuoli, 2014).

Machine learning methods have been widely used for educational contexts in recent years. Decision trees, Bayesian models and other prediction technologies have been applied to address students' admission and counselling process (Ranjan & Khalil, 2008). Hurley et al. trained a decision tree model in order to recommend proper intervention strategies to teachers according to students' profile data (Hurley & Weibelzahl, 2007). A neural network was built to explain students' grades (Gedeon & Turner, 1993). Bayesian networks have been used to predict students' learning performance (Nguyen Thi Ngoc & Haddawy, 2007). Besides, clustering technologies have been widely used in personalized learning environments to support students (Lu et al., 2007; Su, Song, Lin, & Li, 2008).

In this thesis, the group re-composition method demands a machine learning method that can explain group outcome (e.g. high or low learning performance) in terms of combinations of group roles. Although a bunch of supervised learning models can handle such a task, decision tree would be one of the best choices, because it can bring about a concrete structure of the classification and thus is able to yield group composition rules the method demands. On the contrary, neural network and Support Vector Machine (SVM) cannot produce such group composition rules, although they often perform very well in prediction accuracy.

Grouping algorithms have already been closely examined in Chapter 3. Note that "grouping algorithms" and "group composition algorithms" are interchangeable in this thesis. The group composition rules in this case can be considered as the grouping criteria, in the sense that when one inputs a combination of individuals' roles of a group, a predictive outcome of such a group can be told based on those group composition rules. Since those grouping criteria do not relate to peer distance at all, the ant colony optimization and adapted k-means algorithm do not fit this case in spite of their superior grouping quality (cf. Section 3.3.2). Instead, the discrete-PSO and genetic algorithm are immune to such grouping criteria of group composition rules type. The discrete-PSO algorithm is highly suggested here because of its higher grouping quality as seen in Section 3.3.2. Note that such suggestion is based on the evaluation of the selected algorithms in this thesis. Some other grouping algorithms that have not been investigated might also be a good fit.

Again, the dynamic group re-composition method demands support from the recent technologies. Fortunately, as indicated in the analysis above, the recent advancement of such technologies addresses the possibilities of putting the method into practice. Nevertheless, combining all those together in order to fulfil group re-composition is a very novel attempt and thus appears challenging.

4.2 System design

As discussed in Section 4.1.3, the recent MOOCs could be the best-suited application scenario for the proposed group re-composition method. However, many of the current MOOCs do not internally support group work. This demands an external tool that can fulfil the group re-composition goal. For the sake of this reason, a group tool, namely twoleaves, has been developed. The following starts to overview its high-level design.

4.2.1 Requirements modelling

First, to fulfil the goal of group re-composition, the tool must recompose students into new groups when a new collaborative task begins. Basically, there are two options. The first is to forcefully recompose all the participants. However, this would be more or less against some students' will, especially ones from the relatively successful groups. They may know how to effectively work with their current group members and hope to continuously work with them. If facing new members after the process of group re-composition, they may need much more time to know their new members. From the pedagogical perspective, this action certainly hurts their learning experience to some extent. Besides, this evidently violates the spirit of self-directed learning. As a consequence, such drawbacks lead the design in favor of a second option: letting students themselves decide if they stay in the current group or leave for a new one.

Second, as depicted in Figure 4.1, the group re-composition method needs group interaction data, however, few of the current MOOC platforms support intra-group interaction. The designed tool thus requires group communication functionality as well. Up to now, either synchronous or asynchronous communication tools have been widely brought to market. Synchronous interaction emphasizes a timely communication, such as online chatting and audio/video conference, whereas asynchronous interaction (such as discussion forums) is topic-centered in most cases and does not require users to be online at the same time. Recent studies point out that both interaction means have positive impacts on online learning (G. Johnson, 2006; Oztok, Zingaro, Brett, & Hewitt, 2013). Considering the ease of structurally analyzing asynchronous interaction (e.g. online forum board), mainly because of widely-used technologies to analyze forum data (Bratitsis & Dimitracopoulou, 2006; Marcos-García et al., 2015; Marra, Moore, & Klimczak, 2004; Rabbany et al., 2011; A. F. Wise, Zhao, & Hausknecht, 2013), I chose to implement a group discussion forum rather than a synchronous chatting box.

Third, group success, as a second data source, must be given in the process of group re-composition. Group cohesion, group effectiveness and group learning performance can reflect group success from different angles. The selection of which facet depends on what your application tends to emphasize. In twoleaves, the hope is that students on a basis of group can collaboratively solve their teachers' task. Technically, a collaborative writing service needs to be offered for students to collaboratively write down their solutions. To be put simply, the learning performance can thus be derived from their co-writings, for instance, the helpfulness of their solutions and even the goodness of the writing itself (e.g. grammar points, understandability, etc.). Thereby, twoleaves chooses the learning performance as a metric to reflect each group's success. Moreover, we need to address how the learning performance should technically be given. Due to the limitation of scaling up teachers' manual scoring in large online classes, employing a prevalent peer assessment strategy that encourages students to grade their peers' work would rather make such a task actionable (Piech et al., 2013; Vogelsang & Ruppertz, 2015). Overall, co-writing and peer assessment are analyzed to be another two requirements of the developed tool.

If the former addresses the requirements of students, the following should start to analyze the teachers' interests. From the teachers' side, firstly and most importantly, they must assign group tasks. Apart from the tasks' description and timing, the grading criteria should be given as well. With such criteria, teachers themselves can also emphasize some of them, via giving comparatively more weight to certain elements. Secondly, teachers can also join their students' discussions in groups. All groups thus need to be globally accessible to the teachers. Besides, students' writings and peer assessment results might be interesting to their teachers. Thereby, giving teachers accesses to such reveals necessities too.

Overall, Figure 4.3 depicts all basic requirements from both students' and teachers' sides. In addition to these requirements, there could be more if we continue to analyze, for instance, that each student may need a dashboard with statistics of their historical activities (e.g. how many discussion topics made). Such might help with students' self-awareness of their learning. But that is beyond the basic requirements of the group re-composition schema. Thus, in the first version of twoleaves, such functions were excluded and the focus was cast on those minimal requirements we analyzed above.

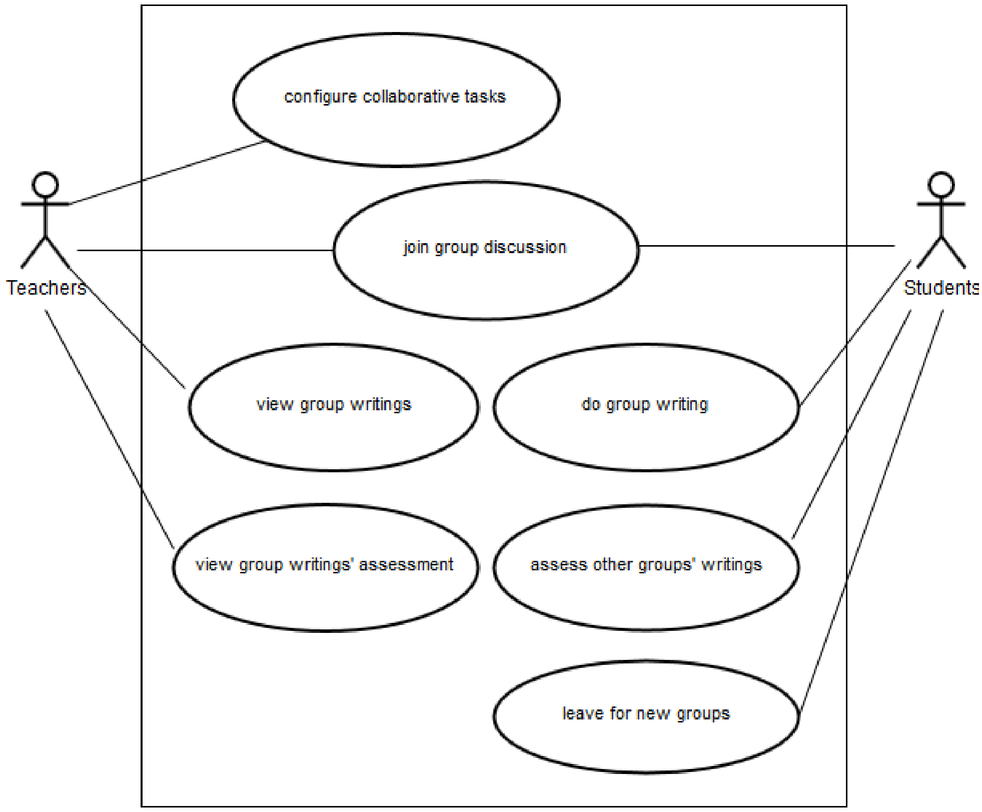


Figure 4.3 use cases of twoleaves

4.2.2 Business logic

Students and teachers are considered to be the two main parties using the twoleaves tool. Students join group discussions, do group writing, assess other groups' works and apply for leaving their current group for a new one. Likewise, teachers need to configure collaborative tasks for their students upfront (cf. Figure 4.3).

A typical business logic of twoleaves is depicted in Figure 4.4. First of all, teachers need to configure a series of group tasks. Each task needs a task name, description of the task (i.e. what students should do), starting date, ending date, peer assessment deadline and peer assessment criteria.

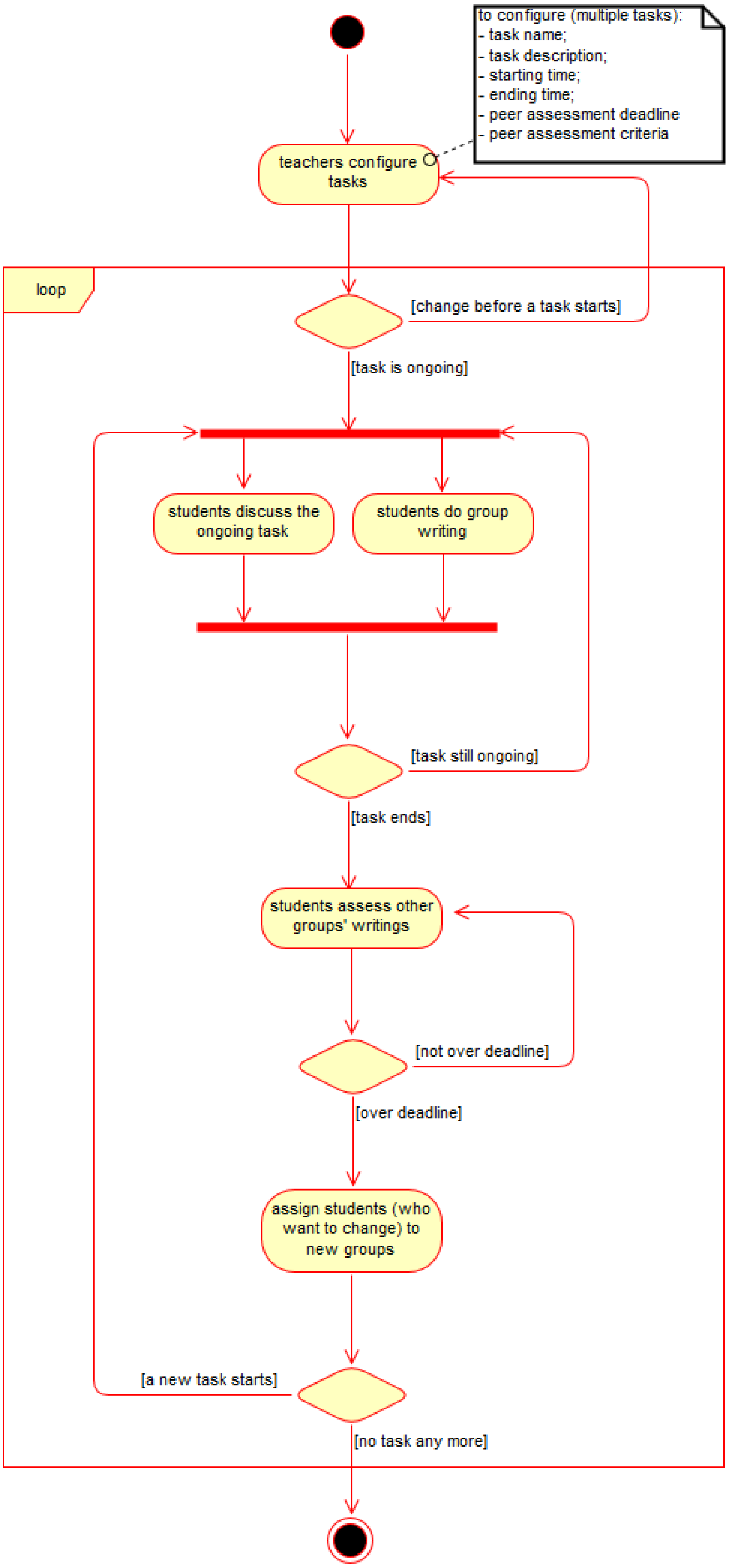


Figure 4.4 activity diagram of twoleaves

When a task starts, students can join group discussions via a text forum where students can create discussion topics and comment on them. Meanwhile, students need to collaboratively complete group writing. Such group writing will be taken as a group solution to the assigned task. This co-writing actually requires many teamwork skills. For instance, students need to come up with a commonly-recognized story line of the writing. Technically, they can achieve this goal in two different ways. First, group members seriously discuss the writing (e.g. what they are going to write down and how to arrange the content) before they write something down. On the contrary, the second strategy could start with their writing. Once they find some conflicted viewpoints or wrong structural arrangement, they could then launch discussions to solve those problems. They may take this strategy many times until they complete their group writing. The nature of this strategy is very analogous to the trial-and-error method in problem solving. Nevertheless, either way relies on the group discussion forum to fulfil group interaction and thus can theoretically encourage more group discussions. However, extreme misuses could probably happen in some groups too. For instance, one single student may do all of the jobs very well and the other members can luckily take a free ride in the end. In such a case, the roles of those free riders could be misinformed because they could have done better than a free-rider if their group was not manipulated by that smart person. We need to keep an eye on how often this happens in real cases.

As soon as a group task is overdue, students are not allowed to edit their group writings any longer and the system in the meantime assigns such writings to other groups' participants other than their own group members for peer assessment. Students can score the works in terms of their teachers' pre-defined criteria.

Once peer assessment is over, the regrouping component of the system starts to calculate each group's overall score, detect each student's role based on their interaction data retrieved from the group text forum and infer group composition rules using roles data and group overall score. The resulting group composition rules are eventually leveraged to suggest new groups for those students who want to leave their current groups (twoleaves provides a user interface for students to express such a will).

After viewing the whole business logic of twoleaves (cf. Figure 4.4), the reminder of this section starts to elaborate the logic of each inner part in detail. The focus will be cast on the five important use cases in the twoleaves system, namely, configure collaborative tasks, join group discussion, do group writing, assess other groups' writings and leave for new groups (cf. Figure 4.3).

4.2.2.1 Configure collaborative tasks

As shown in Figure 4.5, only teachers are allowed to configure group tasks. Aside from the task description and task deadlines, the peer assessment deadline is also very important for twoleaves. Without this, the system would not know when exactly the group success data is ready and thus does not know when to start the group re-composition step. Regarding peer assessment, teachers can define their own peer assessment criteria. Those, for example, can cover grammar points or completion of the work. Teachers can also emphasize some of those criteria by assigning relatively more weight to them.

As long as the task has not begun yet, the teacher can still modify its configuration. In case of emergency, this would reveal its necessity. For instance, if teachers have to postpone the whole course for some reasons, the timing of the group tasks should accordingly be modified.

Once the teacher submits the course's configuration information, the database will update this information accordingly (see details of data model in Section 4.2.3).

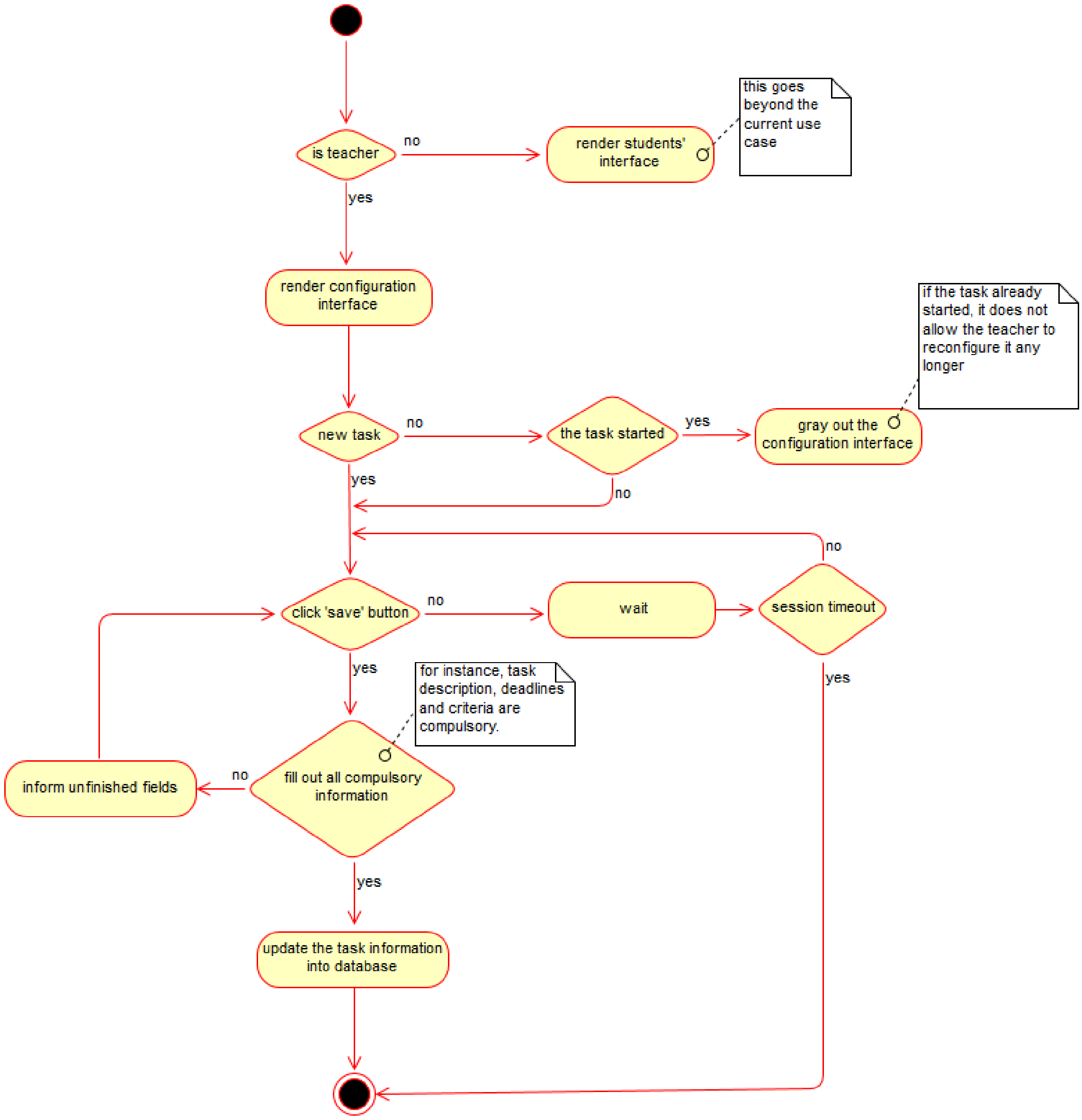


Figure 4.5 Activity diagram of configuring group tasks

4.2.2.2 Join group discussion

As shown in Figure 4.6, teachers and students have different views of the group discussions, although they both can join and participate. To teachers, all students' discussions are open. They can then select some of the discussion topics to follow up. To students, only the discussions from their own groups are accessible. Since each learning group works as an independent unit to conduct learning activities, they are not necessarily allowed to view other groups' discussions.

When either teachers or students make some actions in the discussion forum (e.g. create a new post and reply to some postings), this will accordingly update in the database. In order to cut down on the life cycle of development, twoleaves chose to adapt a reusable third-party discussion forum application, namely django-spirit[15], to meet its needs.

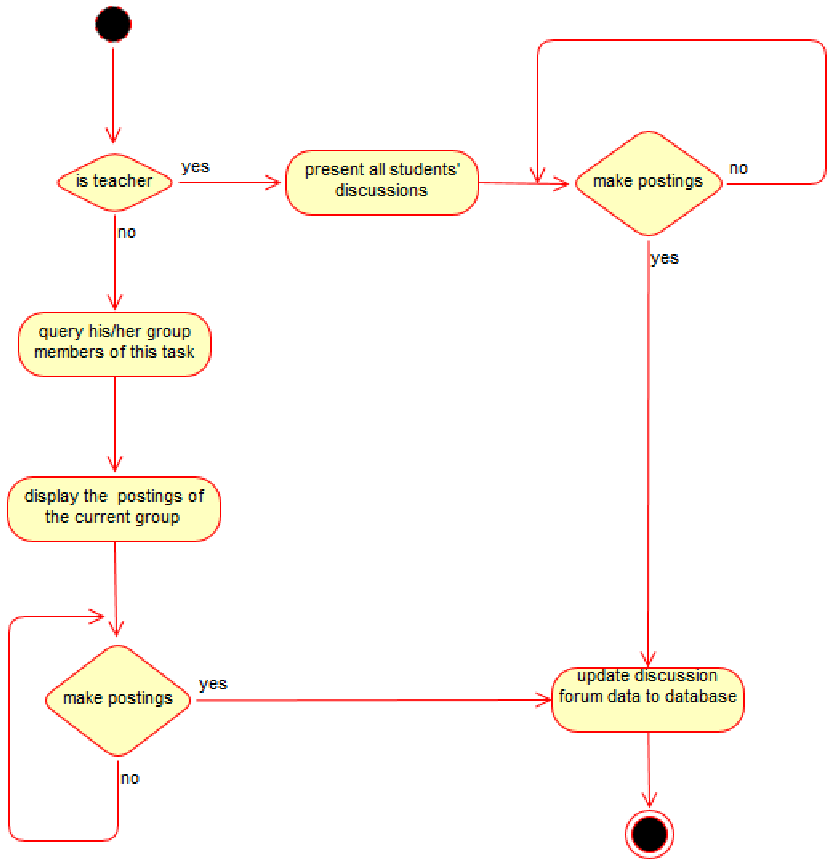


Figure 4.6 Activity diagram of joining group discussion

4.2.2.3 Do group writing

To support collaborative writing, twoleaves integrates a widely-used web application, namely etherpad-lite[16]. This etherpad-lite program basically supports collaborative editing from multiple users in real-time. In twoleaves, students can make use of such a service to work on an online co-writing document. Twoleaves needs to instantiate an etherpad-lite for each learning group when they start to do collaborative writing.

For teachers, they can always view their students' writings but cannot modify anything. As soon as the task ends (referring to task ending time in teachers' configuration), the students are not allowed to edit the task either. The group writings then go to the process of peer grading.

[15] https://pypi.python.org/pypi/django-spirit/

[16] https://github.com/ether/etherpad-lite

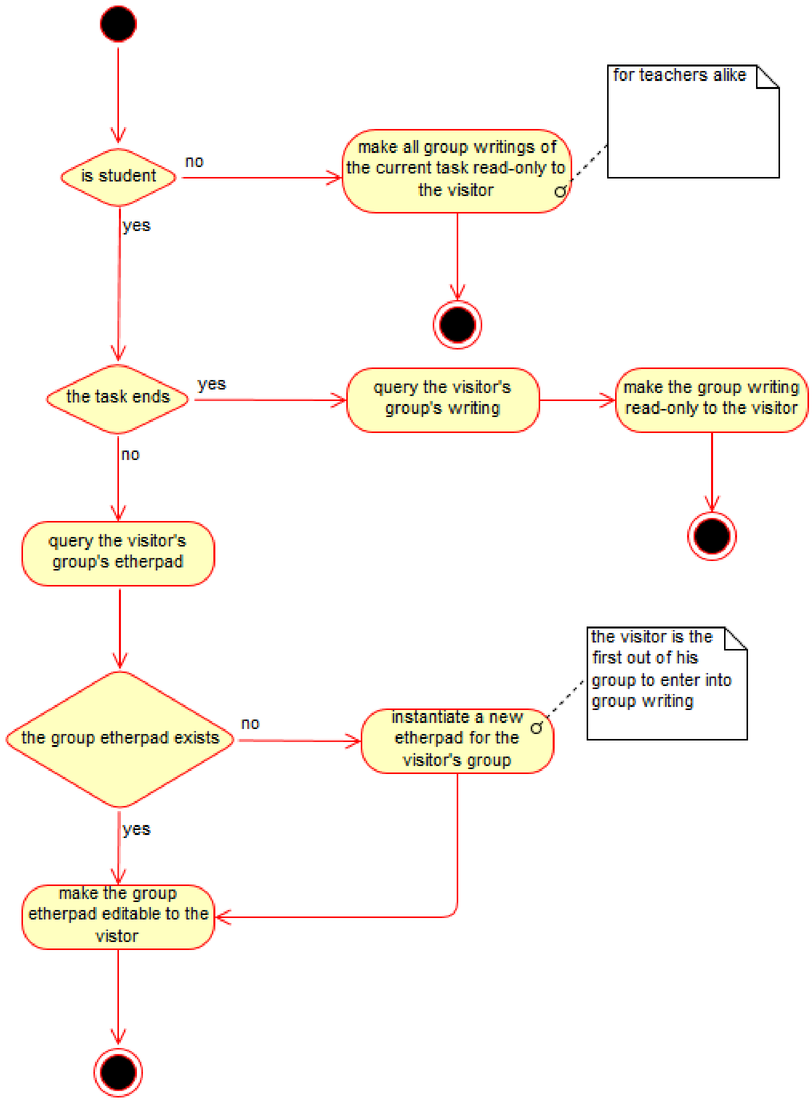


Figure 4.7 Activity diagram of co-writing

4.2.2.4 Assess other groups' writings

Entering into peer grading, teachers and students should have different views. Teachers themselves do not necessarily need to participate in the peer grading procedure and score the works, but they might be intersted to see the scores of their students' group works. For students, they are invited to join such an event as long as it is not overdue. Apart from the assigned group work to assess, the grouping criteria defined by their teachers should display on the same web page. The students assessers simply score the work according to each assessement criterion. When the assessement is submited, the system needs to calculate an overall score for each work based both on the grades given by the student assessers and each criteiron's weight defined by the teacher upfront. The overall score will be used by the group re-composition procedure and will need to be stored in the database.

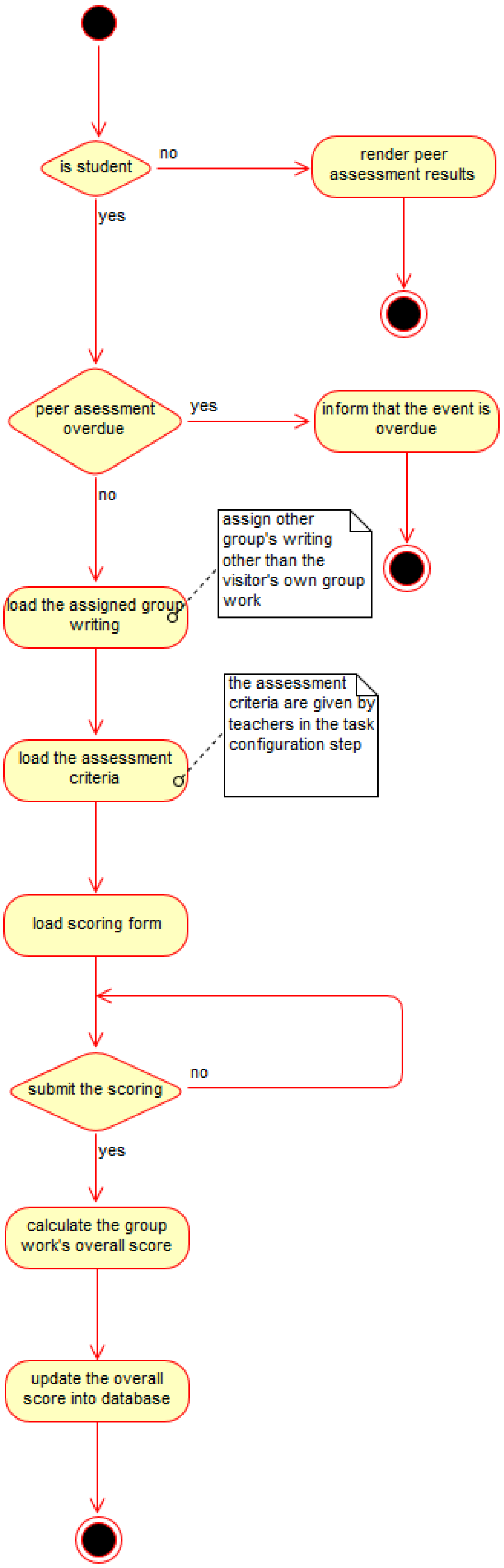


Figure 4.8 Activity diagram of peer assessment

4.2.2.5 Leave for new groups

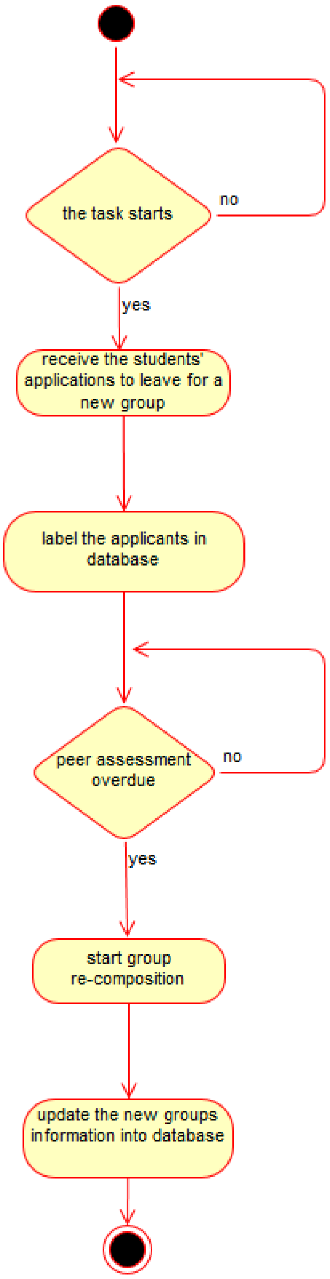


Figure 4.9 Activity diagram of leaving for new groups

The module of leaving for new groups is only visible to students. Once the task begins, the enrolled students can apply for a new group as far as they feel dissatisfied about their current group. When such an application is submitted successfully, the system needs to update a label in the database to mark those applicants. Twoleaves re-composes learning groups merely for those applicants rather than for all participants (see reasons in Section 4.2.1). Thus, those labels are very helpful afterwards. As long as the peer assessment procedure is over, which means the group performance data is ready, the system triggers a group re-composition procedure. This group-composition procedure fulfils implementation of the proposed dynamic group re-composition approach (cf. Section 4.1). After re-composing groups, the involved students should work with their newly assigned group members and the new groups' information has to be updated in the database in the meantime.

Regarding the group re-composition procedure, first of all, a timer starts the whole process when the due date of peer assessment is reached. Next, a roles detector starts to identify each individual student's role using their group interaction data (see roles detector in Section 5.2.1.2) and updates the role data into the database. Then, a group composition learner takes responsibility for inducing the group composition rules using the students' role data and their group performance data (see machine learning component in Section 5.2.1.2). Consequently, the resulting composition rules are used to compose the students who prefer a new group into new groups (see regrouping component in Section 5.2.1.2). Finally, it updates the students' new group information into the database.

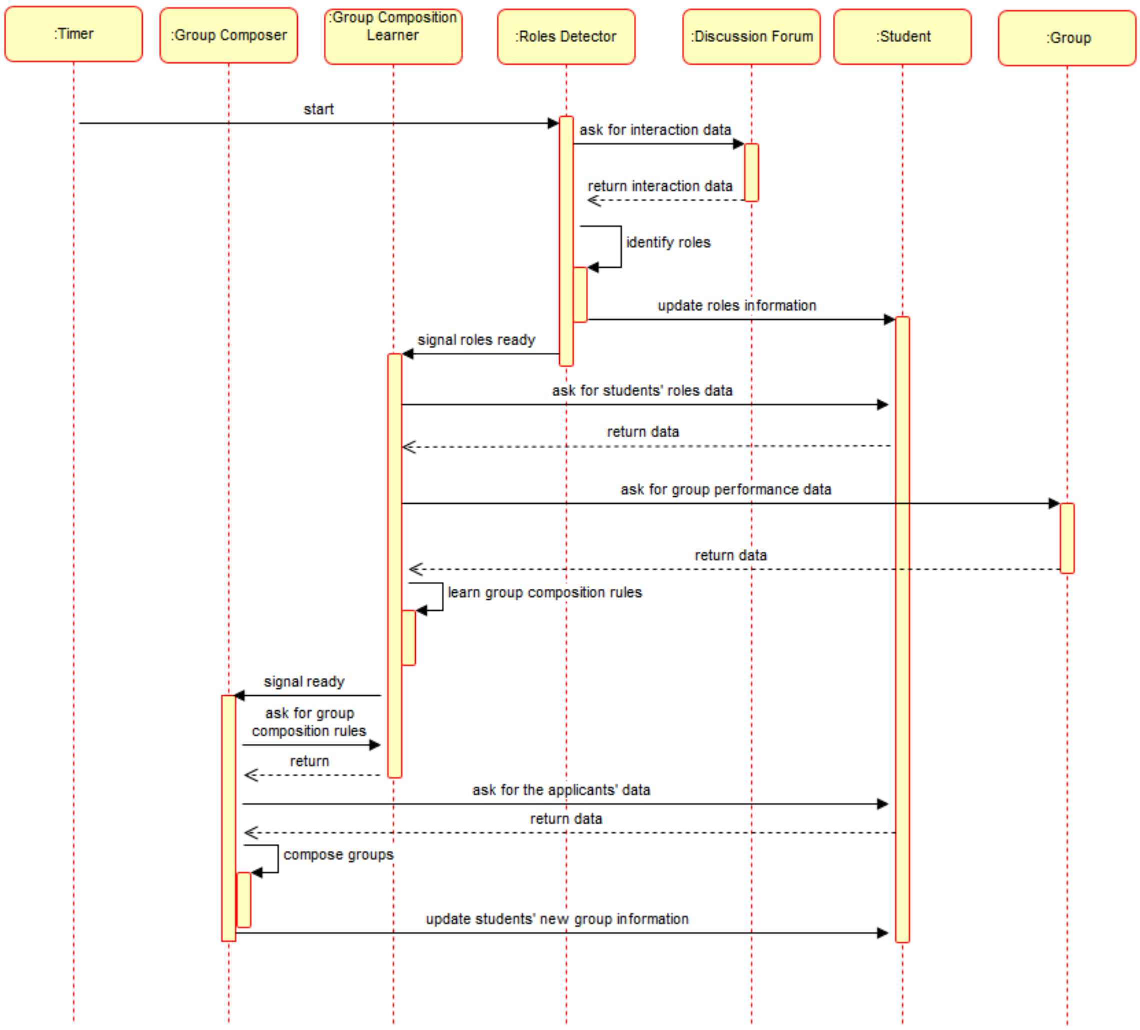


Figure 4.10 Sequence diagram of the group re-composition procedure

4.2.3 Database design

Twoleaves is a data-intensive web application in the sense that it relies on dynamic content (e.g. students' activities in the discussion forum) rather than mere static HTML pages. Thus, it requires a data model to store data and manage relations among those data. When elaborating on the data model, it can, in general, center around three important entities: user, group and task (cf. Figure 4.11)

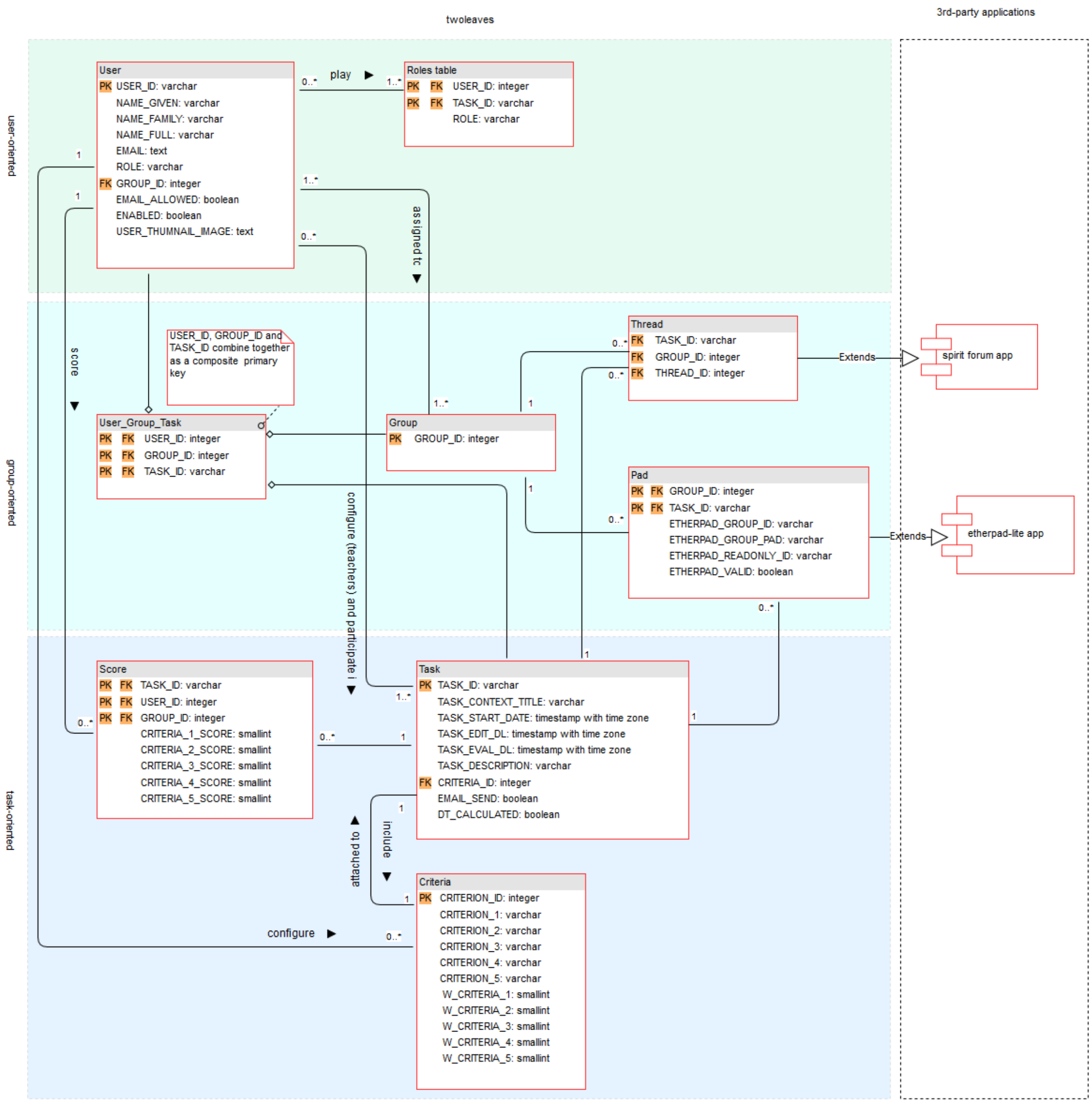


Figure 4.11 Database design diagram

User-oriented data mainly contains students' identity (e.g. user ID, name, email contact and thumbnail image). The user's role as teacher or student is also very important because, as aforementioned, teachers and students should have different user interface to the system. For instance, teachers need to configure group tasks, whereas students do not. These data can be retrieved from the upstream learning platform via a widely-used LTI interface (see Section 4.3.4), which means that twoleaves does not require any user to complete the process of re-registration of his/her personal information any more. GROUP_ID is used to identify the current group of the visiting user. A boolean attribute, EMAIL_ALLOWED, is used to give student the options to select if they would like to receive emails from the system or not. These emails are mainly given to notify students some important events, such as an announcement to start peer assessment. The attribute, ENABLED, is used to flag down the users' entry to the system if they chose to inactivate their accounts in the system beforehand. Aside from these, the roles students play in each task are recorded in the role table. As demonstrated in Section 4.1.1, the roles data is very crucial to the procedure of group re-composition.

Group-oriented data consists of a group table, which is used to reveal a combination of all users, tasks and groups information (namely, User_Group_Task table) and another two tables used to connect groups to group discussion forum and group etherpad respectively. Note that two third-party applications, namely Django-spirit (ver. 0.4.7)[15] and etherpad-lite (ver. 0.2.1)[17], were integrated into twoleaves so as to implement the functionalities of group discussion and collaborative writing. Since the data models of both applications apparently do not belong to the scope of the twoleaves' design, the details are not manifested in the diagram. For more details refer to their source code and descriptions published at github.com (Django-spirit[18] and etherpad-lite[16]).

Task-oriented data incorporates task configurations and scores given to each task. Task configurations contain when to start (i.e. TASK_START_DATE), when to terminate (i.e. TASK_EDIT_DL), when to end peer assessment (i.e. TASK_EVAL_DL) and what task (i.e. TASK_CONTEXT_TITLE and TASK_DESCRIPTION). EMAIL_SEND is used to label if the notification of peer assessment has been already sent out. DT_CALCULATED is used to signal if the decision tree (i.e. the group composition rules) has already been induced from the data. As soon as the decision tree is ready, the group re-composition procedure should begin. Each task needs a set of grading criteria as well. Those are normally defined by

[17] http://etherpad.org/

[18] https://github.com/nitely/Spirit

instructors and then stored in the criteria table. As shown in the criteria table, teachers can define up to five criteria as well as their corresponding weights. Rather than hard coding, it would be better to define a variable number of criteria. Teachers can thus define such according to each task's specific needs. This shows a shortcoming of the current version. An improvement of this will be produced in an upcoming version. The score table stores the scoring data given by peer reviewers. These scores are then used to calculate the group work's overall score, which is of great importance to the group re-composition procedure. Certainly, it is also important to return these data to the upstream learning platform for some reasons (e.g. the learning platform could need to fuse these data into the students' grade book).

4.3 Implementation

Section 4.2 described twoleaves' system requirements, business logic and database design, which attempts to address the system design. The following will outline the technical details and intend to address how to implement such a tool. It will mainly elaborate on the development environment (e.g. web framework and database), deployment, user interface and the integration into MOOCs.

4.3.1 Development environment

Twoleaves was developed under a Python web development framework: Django[19] (version 1.8). Reasons for choosing Django are threefold. First, Django is a free and secured open-source framework. It makes the web development fast because it includes many commonly-needed components of web applications into its core framework (e.g. web caching, logging, sessions and user authentication, see more in online django documentation[20]). Second, Django is written in python, which implies possibilities to reuse a large number of existing python applications. For example, the group discussion function in twoleaves was developed based on a reusable python-powered web application called Django-spirit[15]. Third, the developers (Jan Bundrock and the author) are expert at the python language, which to some extent helped to bring about the decision to choose a python-based web framework.

---

[19] https://www.djangoproject.com/

[20] https://docs.djangoproject.com/en/1.8/

A Django project is a collection of configurations and applications (Azzopardi & Maxwell, 2017). Each application can be considered as a relatively independent component. The philosophy of such a design is to decouple those applications so that each of them can be easily adapted as a component of a different Django project. Such a design is also applied to the twoleaves project. If one glances at the twoleaves' source code (cf. the repository at gitlab[21]), he can find seven applications under the project's directory. As shown in Figure 4.12, each of these structurally separated seven applications has its own configured URL(s). Most importantly, the aforementioned use cases (cf. Section 4.2.2) were implemented by five out of these seven applications respectively. The remaining *lti* application tackles the task of authenticating users' visit from upstream learning environments (see lti connection in Section 4.3.4). Lastly, the *options* application deals with users' requests to deactivate their accounts and reject email notifications.

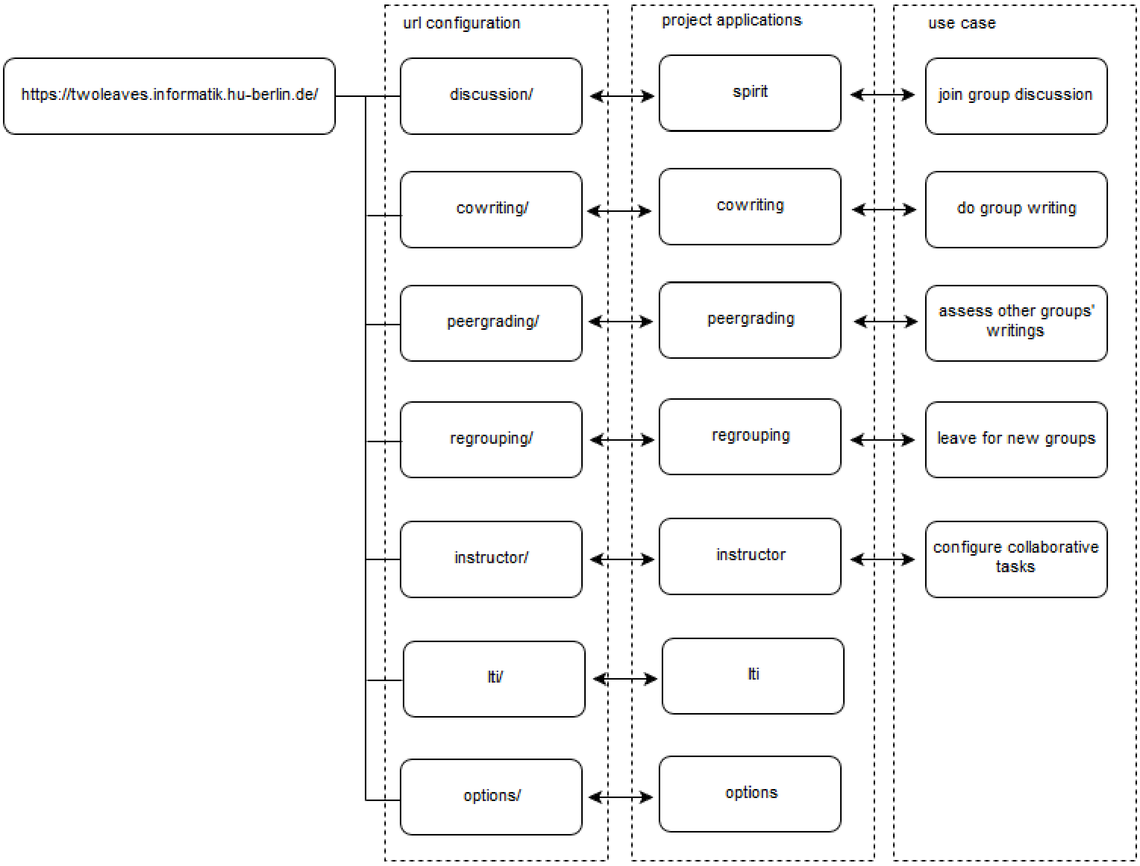


Figure 4.12 twoleaves' project structure

[21] https://repo.cses.informatik.hu-berlin.de/gitlab/zhengzhi/floete

4.3.2 Deployment

As shown in Figure 4.13, twoleaves follows a classic client/server communication pattern. Users interact with twoleaves via their web browsers. An apache2 web server[22] was configured to host the developed Django application. HTTPS (HTTP Secure)[23], was used to secure and mediate the communication between users and the server end. A widely-used open source database, PostgreSQL[24], was selected to store and manage the data needed. Django itself provides a PostgreSQL database interface that can directly operate the database using python language in application and thus brings much convenience to the development.

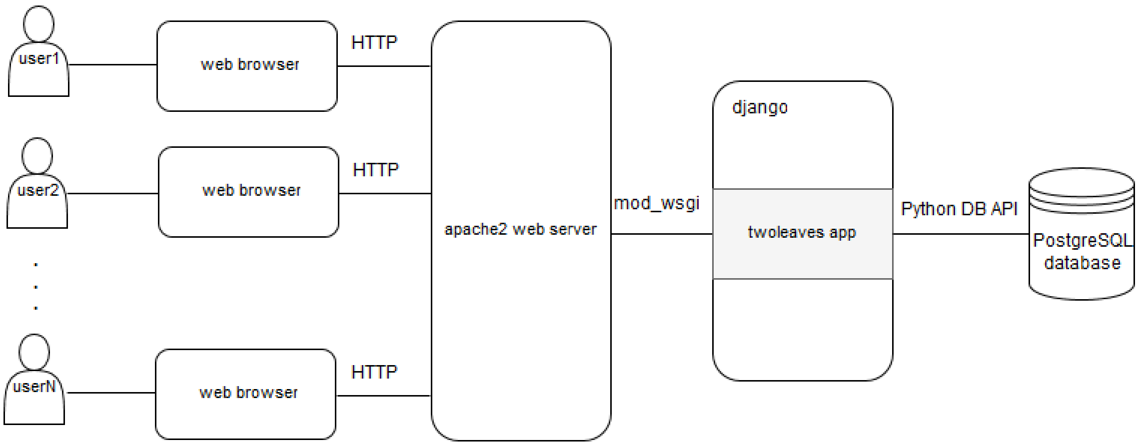


Figure 4.13 Twoleaves deployment diagram

4.3.3 User Interface

User interface is very important for users to interact with the learning tool. The following attempts to navigate you throughout the main functionalities of the twoleaves system: task configuration, group discussion, co-writing, peer assessment and regrouping. After having read Section 4.2.2, one may find that they strictly reflect the five main use cases of twoleaves.

Task configuration is designed merely for teacher users. When a teacher successfully logs into twoleaves via an upstream learning platform (e.g. MOOC), he/she can basically see two main fields: task information and assessment criteria (cf. Figure

[22] https://httpd.apache.org/

[23] https://tools.ietf.org/html/rfc2818

[24] https://www.postgresql.org/

4.14). Task information mainly contains the task description and its life circle. The following explains each item.

- *task name*: a name of the task.
- *task description*: a field to specify what exactly you want students to do in the task.
- *starting time*: date when the task starts.
- *ending time*: date when the task stops (specifically, no allowance of co-writing on the task any more).
- *peer assessment deadline*: date when to terminate peer assesment on the task. As soon as the task ends (refering to ending time), the system starts to organize peer assessment. Thus, this dealine has to be set later than the ending time.

With regard to assessment criteria, teachers can change the default criteria to meet their own needs. Each criterion can be weighted (range from 1 to 10) in terms of its importance.

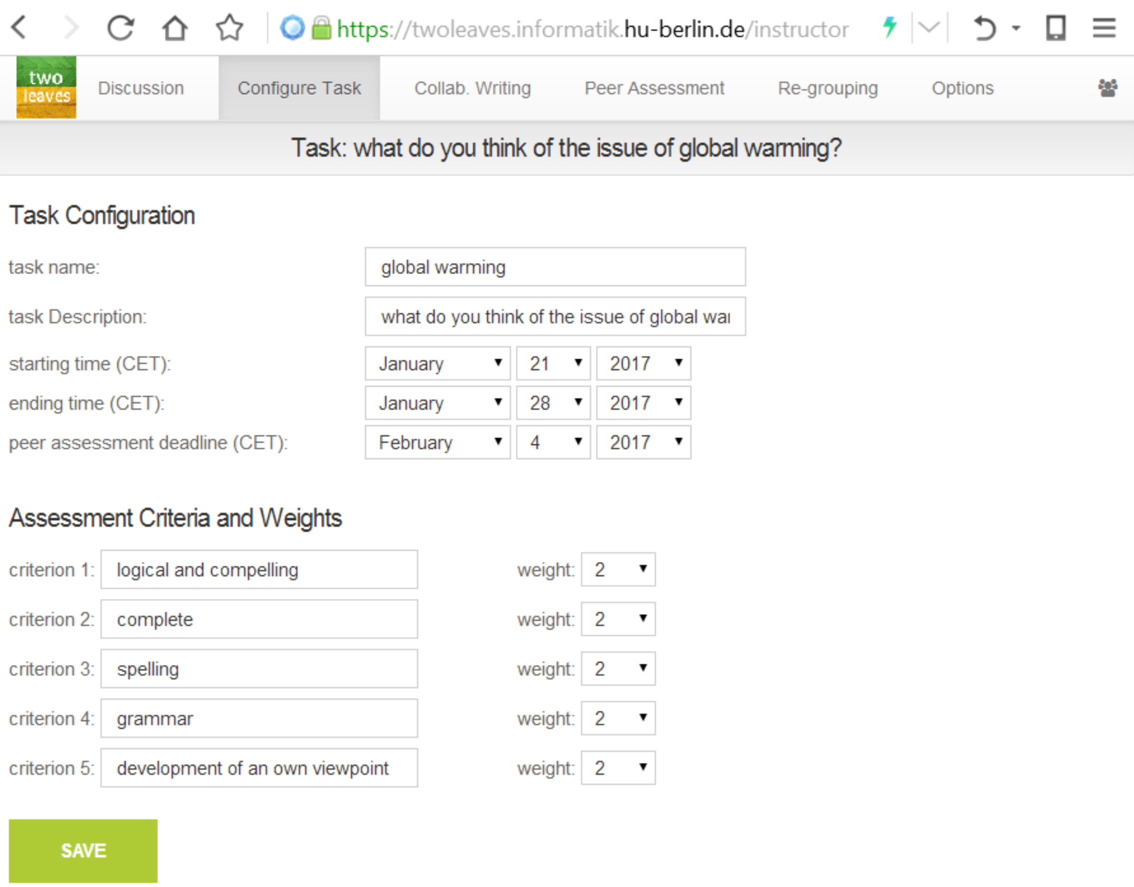


Figure 4.14 User interface of instructor's configuration

Group discussion is fulfilled via an embedded discussion board (cf. Figure 4.15). It provides with the features that can be easily seen in up-to-date online discussion boards (see features on spirit-forum project[25]). Through this communication channel, students can create discussion topics, ask and answer questions, share their ideas on the discussion topics and even organize and coordinate their co-writing work. Different from the general course forum in MOOCs, this forum is rather visible to group members other than the whole audience of the course.

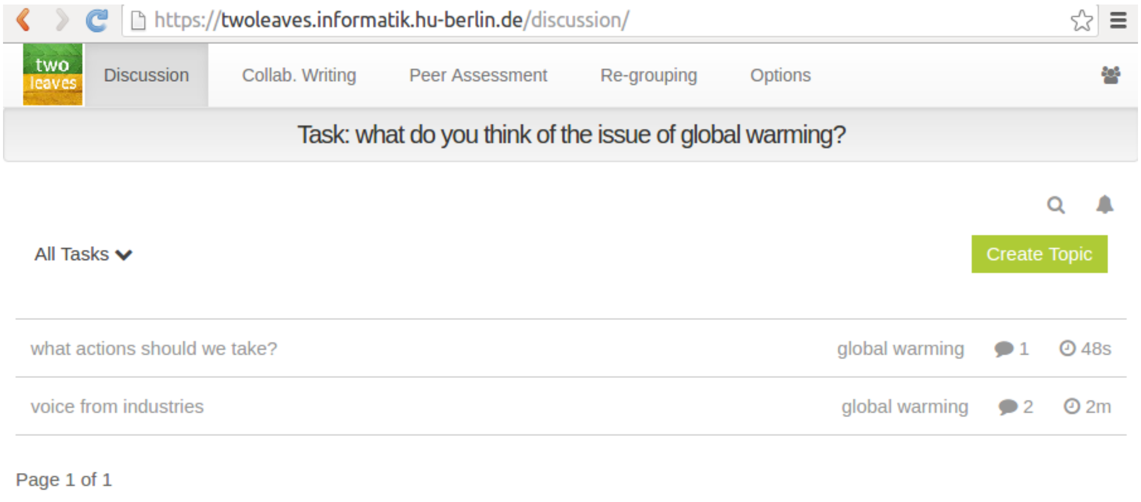


Figure 4.15 User interface of group discussion

Co-writing facilitates a collaborative space for group members to work on their solutions to the assigned task. To implement this, an up-to-date etherpad was integrated (cf. Figure 4.16). It allows multiple users to edit the document simultaneously. The writing of each participate is visible to all group members.

[25] http://community.spirit-project.com/

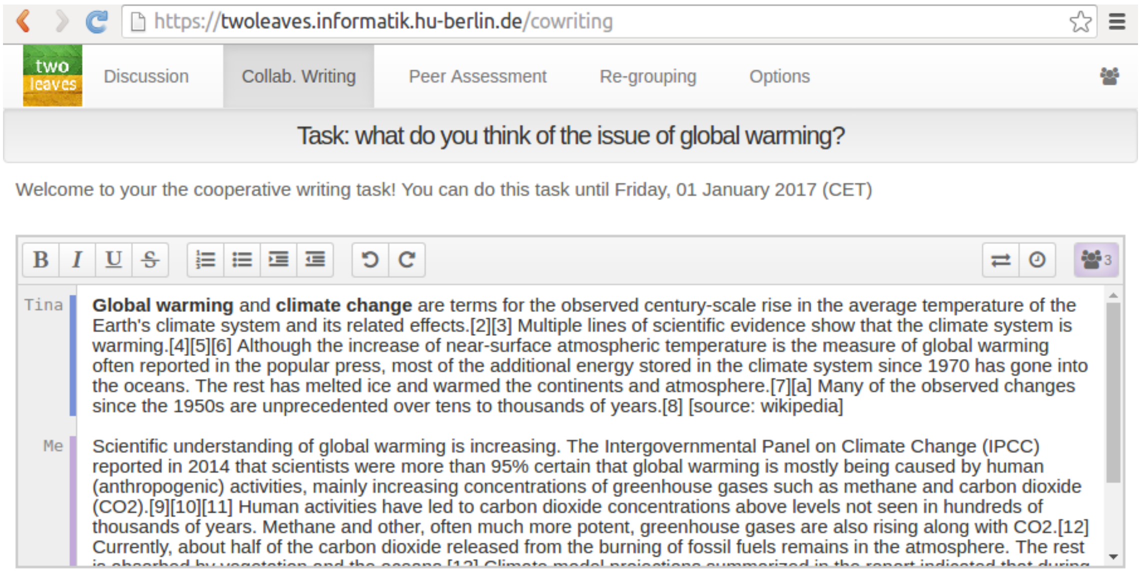


Figure 4.16 User interface of collaborative writing

Peer assessment is due to start as soon as co-writing finalizes. We invite the task participants to review and rate the group works. By default, each student is invited to rate at least one group work. For convenience, the teacher-defined assessment criteria are appended to the assigned review work (see Figure 4.17). Student reviewers only need to give their scores according to each criterion. The system will calculate the overall score of the work according to the scoring and the criteria's weights. Note that the system does not allow students to assess their own group's work.

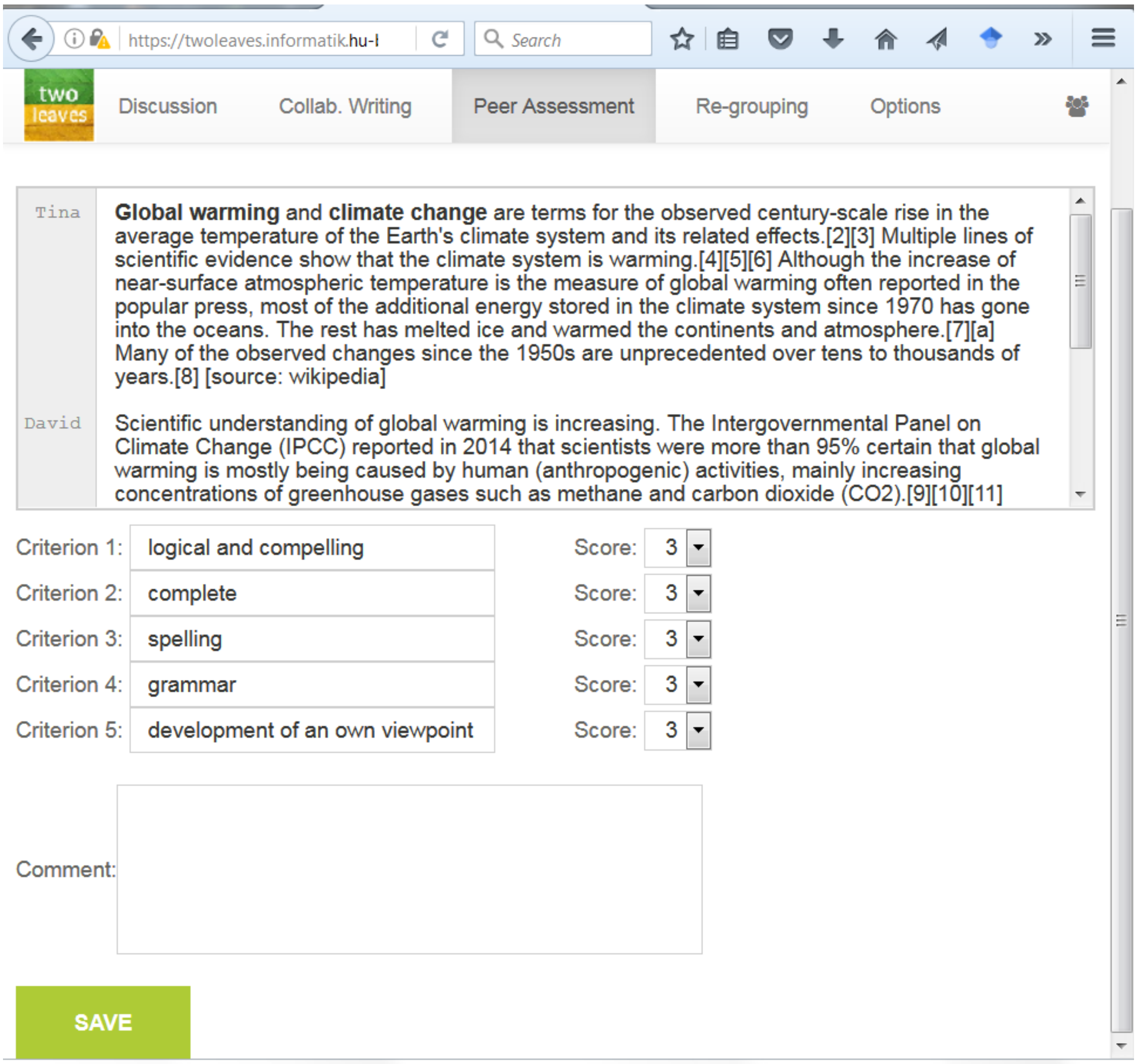


Figure 4.17 User interface of peer assessment

Regrouping undoutedly addresses the problem of group re-composition. If some of the participants want to leave their current group for a new one, they can express such a will in twoleaves simply by clicking a button (cf. Figure 4.18). Note that new groups will come into effect at the start of the next task. Apart from this, participants can also give feedback about their reasons for leaving, so as to unveil more details on their motivations.

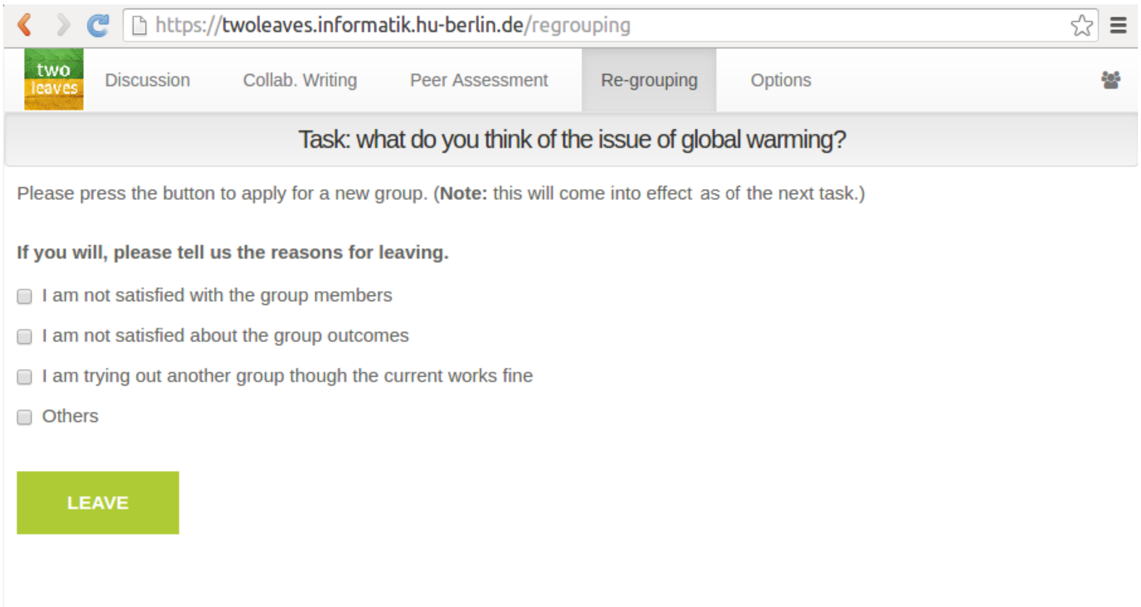


Figure 4.18 User interface of applying for a new group

4.3.4 Integration into MOOCs

User experience is always one of the biggest concerns of online systems. The ease of using small learning tools on large online platforms attracts much attention of web developers. At least, students should not necessarily feel the embedded tool is from a third party other than the learning platform. The following elaboration seeks to address such concern.

4.3.4.1 LTI connection

Twoleaves supports Learning Tools Interoperability®(LTI) standard[26] (version 1.1) with which it can be integrated into many of the current MOOC platforms. LTI was proposed to seamlessly connect a growing number of learning applications (e.g. chat tools) to learning platforms (e.g. MOOCs and learning management systems).

As shown in Figure 4.19, when the MOOC platform launches a LTI connection, the data bound with it will be passed from the upstream platform to twoleaves. This data mainly includes users' profile and learning context information (cf. Table 4.1). Those are very important for twoleaves to identify incoming users (who they are) and the specific learning context they are engaging in, such as user id, user name and learning context identifiers. Note that other parameters aside from those are transmitted (e.g.

[26] http://www.imsglobal.org/activity/learning-tools-interoperability

user's thumbnail image and credential information between the learning platform and twoleaves) as well. A full list of such data is given on LTI's official website[27].

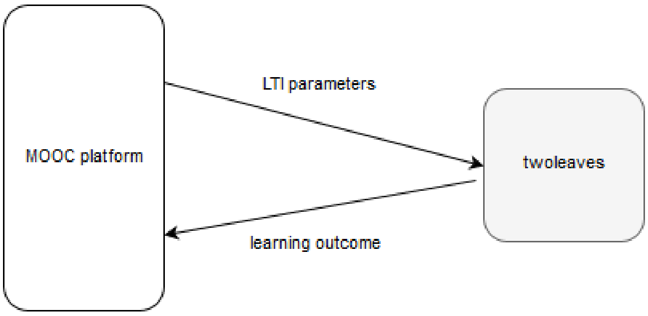


Figure 4.19 Connection between twoleaves and MOOC platform

Table 4.1 LTI parameters that used to identify users and learning contexts

LTI parameters	Description
user_id	an unique identifier for an user
Roles	e.g. student, instructor and so on
lis_person_name_given	User's given name
lis_person_name_family	User's family name
lis_person_name_full	User's full name
lis_person_contact_email_primary	User's email contact
context_id	Uniquely identifies the learning context
context_title	A plain text title of the context
context_label	A plain text label of the context
resource_link_id	Uniquely identifies the resource, each of multiple placement in the same context should be distinct.
resource_link_title	A plain text title of the resource

[27] http://www.imsglobal.org/specs/ltiv1p1/implementation-guide

The learning outcomes that students make in twoleaves might be interesting to the upstream learning platforms as well. They could be collected into their grade book or used to update students' learning progress or even both. In LTI 1.1, there is a parameter provided to fulfil this goal. By definition, however, it merely supports a decimal score ranging from 0 to 1.

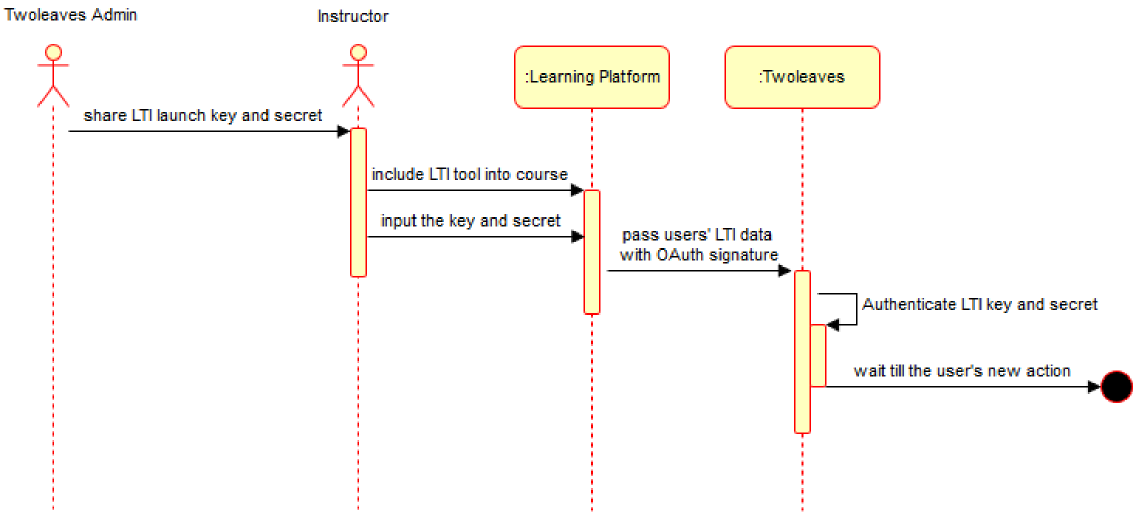


Figure 4.20 Sequence diagram of launching twoleaves on MOOC platform

4.3.4.2 Launch twoleaves in MOOCs

As informed in LTI 1.1's specification[27], there are many ways in which it can be adapted to launch a LTI tool on the upstream learning platform. In general, they can be categorized into two categories. First, the learning platform specifically implements an interface so that the learning platform and the embedded LTI tool can share credentials. Instructors can then simply select the tool and include it in their under-designing learning contents. The second way is to share credentials between instructors and the embedded tool. When instructors demand a LTI tool, they need to manually type in the selected tool's credential information so that the tool can be successfully launched. Because our industry partner's MOOC platform (i.e. iversity) supports the second, twoleaves thereby follows that as well. As shown in Figure 4.20, twoleaves firstly shares the credentials (i.e. a pair of key and secret) with the instructor. The instructor then includes a general LTI interface directly from the learning platform and types in the key and secret. Next, the learning platform passes on the credentials that the instructor provided together with other LTI data to twoleaves. When receiving the upstream learning platform's request, twoleaves, first of all, authenticates the coming request with the shared key and secret. If it is authenticated successfully, the visitor can start working with twoleaves. Otherwise, the visitor would be banned from entering into twoleaves.

With regard to twoleaves' presentation in the upstream learning platforms, there are two options. First, twoleaves can locate in an iframe[28] (an element of HTML). Second, it can be launched in a new browser window. The selection of these two options depends on the platform's settings. Some platforms may worry that their users' learning attention might be distracted if a new browser window pops up as a response. Thus, they prefer to use iframe and embed the tool into the same web page as their learning contents. On the other hand, iframe limits the tool in a very small presentation space and thus could hurt its usability.

4.4 Chapter summary

This chapter introduced a new concept: *dynamic group re-composition* in the field of group formation. When one creates small learning groups in recent big online classes (e.g. MOOCs), the high drop-out rate would result in many groups incomplete in size. The scarcity of left behind human resources in those groups was argued to be an important incentive to come up with this dynamic group re-composition method. This can be considered an objective reason in the sense that we have to do that if we still want those involved groups to continue with their work. A subjective incentive could derive from humans' inherent motive for success. When students are not satisfied with their current group's outcomes and there is an opportunity to leave, why not take it?

The dynamic group re-composition essentially runs on top of the group composition module. After all, it needs that module to make groups in the end. However, this new group re-composition method can mitigate some of the problems that the existing group composition methods could come across. For instance, it widens the possibility to collect and leverage data along group process instead of mere reliance on the static data collected before group process. It can also reflect data dynamics that the previous group composition methods have rarely accounted for.

The proposed dynamic group re-composition method is data-driven. In general, the nature of a data-driven method requires a large amount of data. The current MOOCs

[28] http://www.w3schools.com/tags/tag_iframe.asp

with tens of thousands of students provide this possibility. Besides, in recent years, the wider adoption of social network analysis and machine learning methods in the field of education somehow addresses the related technical worries. In principle, implementing such a method is not unrealistic in this day and age anymore.

A group tool, twoleaves, has been developed in order to fulfil the goals of the proposed dynamic group re-composition method. It can be seamlessly integrated into many of the modern online learning environments with a LTI standardized interface. With twoleaves, students can voluntarily apply for a new group. The twoleaves tool is then responsible for finding a suitable learning group for those applicants on the basis of an automated data analysis mechanism. Besides, group discussion and collaborative writing is well designed to support students' group work too.

5 GROUP RE-COMPOSITION: EVALUATION

Both empirical and simulation methods can be possibly considered as methodologies to evaluate the proposed group re-composition approach. Empirical methods emphasize the way of doing science by means of direct and indirect observation or experience. They usually gain scientific insights through data collection and data analysis (Kinghorn, 2013). Because the empirical observation arises directly from reality, it certainly indicates their inherent strength of best reflecting the real world. However, they bring many worries as well. First, they involve so much human resource and time cost that they are difficult and costly to be implemented in many cases (Dooley, 2002). Second, missing data is always the case in reality (Axelrod, 1997; Davis, Eisenhardt, & Bingham, 2007). The data source could be cut off because of various unanticipated reasons, such as, organizational changes and technical problems. Third, many uncontrollable factors could influence the findings either negatively or positively (Axelrod, 1997). For example, when we study the relationship between students' behaviours and their performance in an online math system, we cannot simply disregard their teachers' interventions in schools that are hardly to be noticed and measured yet (Zheng, Stapel, & Pinkwart, 2016). Fourth, the ethical tension sometimes, if not oftentimes, causes a great deal of concern in empirical studies, such as testing new drugs in medical areas (Ziv, Wolpe, Small, & Glick, 2006). Partly due to reality's great complexity, the weak specification of boundary conditions is another pitfall of empirical methods (Davis et al., 2007).

Table 5.1 Pros and cons of empirical methods

Pros	Cons
+ arise from reality	- costly and difficult - missing data - many uncontrolled factors - ethical tension - weak specification of boundary conditions

Simulation, as another method of doing science, also generates data for inductive analysis as empirical methods do. Simulation data is, however, produced based on assumptions rather than collected from the real world (Axelrod, 1997). Nevertheless, its wide usage reveals its imperatives as well. First, the purity and clarity of data is preserved because of the specified modelling rules and ease of removing uncontrollable variables' influence (Axelrod, 1997; Davis et al., 2007). Second, you can replicate the experiments as many times as possible. This is of vital importance when you want to re-observe a variable's impact under the completely same experimental conditions. In contrast, in an empirical study, this is almost impossible (Kinghorn, 2013). A third benefit is the ease of data collection. You can collect as much data as you need and missing data never occurs in simulation (Axelrod, 1997). Fourth, a systematic study on the effect of changing the parameters is very convenient when conducted. The last point is its clarity of boundary conditions, which benefits from its rigorously defined assumptions and parameterized variables. Nevertheless, critics of simulation point to realism because many simulation models incorporate more or less unrealistic assumptions (Kinghorn, 2013). Regarding this argument, two points have to be made clear. First, simulation is not induction, which means that the assumptions of simulation modelling do not necessarily need to cover every detail of the real world. Second, the accurate representation of the real world is one of the important metrics to measure the quality of simulation but it is not always the case. Having a Look at Axelrod's elaboration: " … *if a simulation is used to train the crew of a supertanker, or to develop tactics for a new fighter aircraft, accuracy is important and simplicity of the model is not. But if the goal is to deepen our understanding of some fundamental process, then simplicity of the assumptions is important and realistic* *representation of all the details of a particular setting is not*." (Axelrod, 1997, p. 6), we might have a better understanding of simulation.

Table 5.2 Pros and cons of simulation methods

Pros	Cons
+ purity and clarity of data + ease of replicating experiments at will + ease of data collection + ability to systematically study the effect of changing the parameters + clarity of boundary conditions	- strip away realism more or less

Back to this thesis, one can make use of the tool, twoleaves (cf. Section 4.2 and 4.3), to conduct an empirical experiment on a real MOOC. However, I have, thus far, not found a suitable online course on our partner's platform (iversity). The reasons could be twofold. First, the requirements of the group re-composition might be very strict. It requires multiple group tasks along a course timeline. This, in fact, requires much more course design work from the teachers' side. Second, teachers could worry about the negative effects of such a new experiment. For instance, students' learning experience could be hurt by integrating a third-party tool into the MOOC platform.

Otherwise, can one evaluate the group re-composition approach via a simulation method? Before answering this question, let us have a look at the goals of this evaluation work. The first goal is to validate the technical feasibility of the proposed approach (e.g. do roles detection and group composition rules induction component function in a proper way?). To achieve this, the key is to seek a group interaction data source as input data. Merely for this goal, the validity of such input data is not of much importance. The data can be obtained either empirically or artificially. As asked in the research question RQ4, the second goal is to examine the benefits that the group re-composition method could bring. This respect undoubtedly needs a valid data source. If the data does not reflect typical group interaction in real groups, the results could accordingly fail to conform to the fact of life. In a word, if we can find a valid group interaction data source, we can say yes to the question of whether simulation would work or not. Thanks to Nygren's works, modelling of online discussion groups via simulation has already been developed and validated (Nygren, 2010, 2012). Such modelling, to some extent, can help to simulate the group interaction and thus solve our data source problem.

All in all, a simulating experiment rather than an empirical study was chosen in this thesis. Nevertheless, it does not literally mean that simulation is superior to an empirical study. Rather, both of them are of the same importance. They could even be done in parallel. However, the lack of a suitable MOOC course impelled me to choose simulation.

This chapter starts with simulation works that have been previously done in educational fields. It then details the simulation experiment. Next it presents the results and finally summarizes the findings.

5.1 Simulation work in educational research

Plenty of researchers have chosen simulation already. AutoTutor, simulating human tutor, was developed to assist college students in learning several of Computer Science-related courses (Graesser, Wiemer-Hastings, Wiemer-Hastings, & Kreuz, 1999). Likewise, Matsuda et al. developed a machine-learning agent, SimStudent, for an Intelligent Tutoring System (ITS). Using SimStudent, they successfully simulated students' steps to solve algebra problems in a cognitive tutor (Matsuda, Cohen, Sewall, Lacerda, & Koedinger, 2007a, 2007b). With regard to simulation of social interaction, Stasser simulated group discussion with an attempt to investigate group decision making (Stasser, 1988); John Gullahorn and Jeanne Gullahorn tried to emphasize computer simulation's active functions for the development of theory via simulating human interaction in small groups (Gullahorn & Gullahorn, 1964); more recently, Padilha and Carletta proposed simulation of small group discussion in order to investigate such nonverbal behaviors as gaze, posture, gesture and facial expression's contributions to the turn-taking process (E. Padilha & Carletta, 2002; E. G. Padilha, 2006); in addtion, online users' participation and interaction in discussion groups has also been modelled by Nygren via their simulation experiments (Nygren, 2010, 2011, 2012).

5.2 Simulating experiment

The following subsections detail a simulation experiment that aims at evaluating the proposed group composition approach. It consists of an overview of the whole simulation process, input, output and elaboration of every component embodied.

5.2.1 Simulation System

5.2.1.1 Overview

The simulation system developed in this thesis is intended to compose students into small learning groups, simulate students' interaction in groups, detect students' roles from their interaction data, induce group composition rules based on students' roles and group success and recompose students into new groups according to the induced the group composition rules. Specifically, it should consist of the following functions:

- A group composer that composes students into initial groups
- A group interaction simulator
- A roles detector that infers each individual's role based on the group interation data
- A machine learning component that inputs group roles data and group cohesion data and trains such a classifier that judges successful or weak group for new group composition
- A regrouping component that offers functions of reforming groups and optimizing group formations iteratively

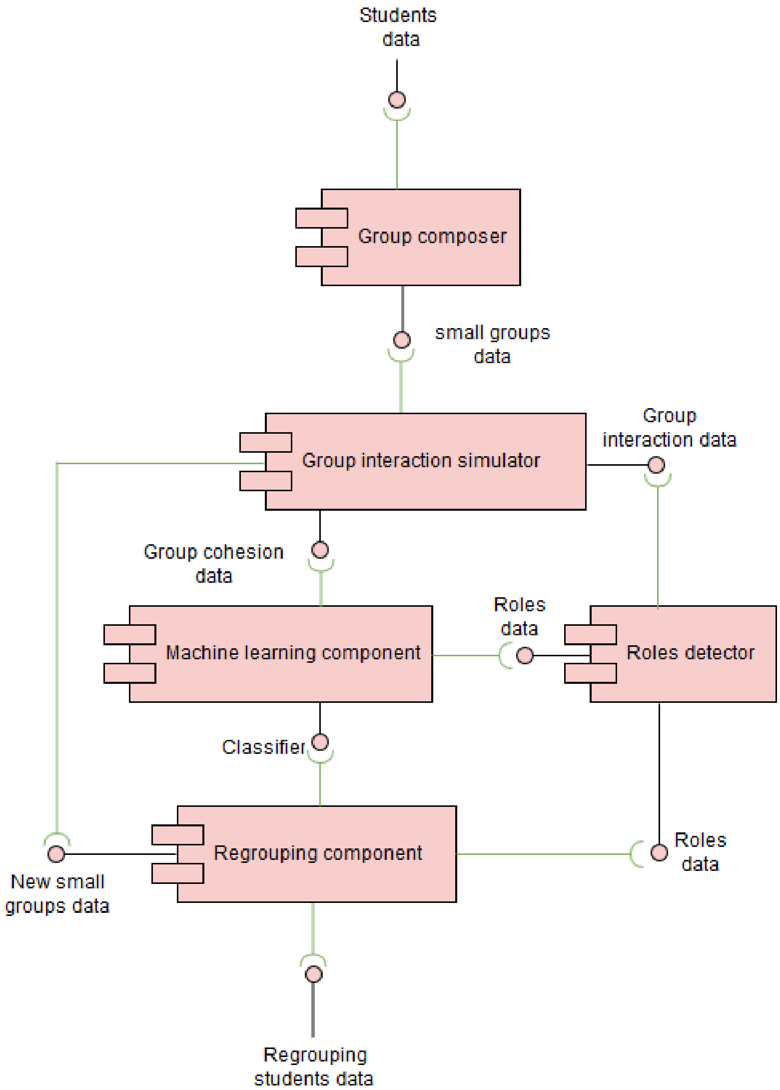


Figure 5.1 Simulation system diagram

As shown in Figure 5.1, a set of students are input into a group composer and small learning groups are yielded. A group interaction simulator then simulates the group interaction of each small group and produces group interaction data simultaneously. With this group interaction data, a roles detector can detect each student's role (e.g. lurkers and leaders) and transfer the data to a machine learning component and a regrouping component. Except for the students' roles data, the machine learning component needs another data source, group cohesion data (why group cohesion instead of learning performance, see Section 5.2.2), derived from the group interaction. With both data sources, the machine learning component induces group composition rules. Those composition rules are basically a classifier that tells us what roles, when combined into a group, yield higher or lower group cohesion. When the classifier is ready, a regrouping component can start to recompose students into new groups via consulting the classifier. After all these, students start their new task in the assigned new groups. How each of those component works will be elaborated in Section 5.2.1.2.

5.2.1.2 Components

Group composer implements the function of composing initial learning groups. It takes a set of students and group size as input. The output of this component is groups of the given group size. Prior to this first step none of students' data is collected (cf. the proposed group re-composition approach in Section 4.1.1). The initial groups can thus only be composed at random. Note that in case of an uneven split over all resulting groups (i.e. the last group could not have the size of the given number), the last composed group can either stay alone or merge into the second to last group. It depends on if the actual size of the group is over half of the given group size. The composition algorithm is shown in Figure 5.2.

```
Input:
       - n // the number of students
       - m // group size

Output:
       - G = [g_1, g_2, ...]
       // a set of small groups

randomly generate a set of n students S
while S is not empty:
       retrieve the first m elements from S as a small group g
if the size of the last group is less than m/2:
       merge it into the last second group and remove the last one
combine all small groups g into G
return G
```

Figure 5.2 Composing initial random groups

Group interaction simulator generates the interaction data that is crucial to proceed to the next steps of the whole simulation. It does not come to the semantic dimension (i.e. topic of the talking) but focuses on social connections among group members. The heart of this component stems from Nygren's work on simulation of user participation and interaction in online discussion groups (Nygren, 2012). Nygren's work, to some extent, lays the foundation for modelling of the group interaction and can so far be recognized as the only one that presents guidance to every detail of such a simulation system. Apparently, that is the reason why I employed and extended it to this thesis. Aiming at showcasing the social structure of group members, such a simulation system principally has to answer the following questions.

- Who will start a discussion post at the next moment?

- Will this post mention other participants (replies, comments and mentions of any sort)?
- If so, whom exactly will be mentioned?

These questions are addressed by introducing two concepts, namely, “the Edge Initiation Process” and “the Edge Destination Selection Process”. The Edge Initiation Process determines who will start the next post. It depends on the grooms participants have received. Here ‘groom’ is a concept popular in social animals which is analogous to the attention we often pay in human interaction, such as nodding in face-to-face scenarios or comments akin in social media tools (Nygren, 2011) (p38). According to Nygren’s studies, the participants whose first post was groomed have more probability to make a second post than the ungroomed ones. And the probability to make a new post is proportional to the accumulative number of posts that have been already made. In this process there is another task that has to be done at the mean time, which is to determine whether this newly-made post contains a groom to others or not. They used a probability parameter which observed from an empirical study to decide on the attachment of a groom. One might ask if men who have not made any post yet are allowed to make a new post in the group. Fortunately yes, another process called “the Node arrival Process” tackles such a task that new participants can join the on-going discussion as well. Basically, every new post could be probabilistically owned by new participants. The probability can be simply given as the number of group participants divided by the total number of posts expected in the group. Obviously, the Edge Initiation Process answers the first two questions.

The Edge Destination Selection Process describes how to select a person who will be groomed if a groom is attached to the post, which tries to address the third question as formulated beforehand. A naïve, yet fair, strategy to be adopted could be to assume that all participants have the same probability to be selected as the destination of grooming. However, this strategy dismisses any association to contributions the participants have already made. That is, no matter how many posts they have made or grooms they have received, the selection is sort of random. In fact, this does not look fair to diligent contributors. In terms of previous empirical studies (Leskovec, Backstrom, Kumar, & Tomkins, 2008; Nygren, 2012), the number of grooms currently received can explain their future potential to be groomed. Literally, the more often the participants are groomed, the higher their chances are to receive a groom again during the following interaction. Another observation indicates that the groom-balance matters as well (Nygren, 2012). The groom-balance is a metric to measure the difference between the number of grooms a participant has given away and the number of grooms he/she has received. Overall, the selection of grooming destination resorts to two strategies: 1) Groom-sum; 2) Groom-balance. Probably, there could be some other factors which are interesting to be investigated, for example, wit-value as mentioned in (Nygren, 2012). Wit-value is intended to model participants' disparities on wit which could relate to one's talent, educational background and even professional expertise. Especially, in the scope of personalized learning, this difference has been seen by many researchers and thus cannot be neglected in some cases. For the sake of its weak impact in the reported experiment, the thesis does not integrate it. Note that this cannot be simply interpreted as its little impact in learning groups. In some application contexts, for example, professional development courses, this factor might exert much more influence than in arts courses for instance. Overall, Figure 5.3 depicts the inner mechanism of the group interaction simulator.

```
paramters:
  - P_nys //Probability to select a not-yet-spoken member as a new speaker
  - P_groom_sum //probability to select a speaker according to the sum of
grooms received already
  - P_groomed //probability to select a speaker out of groomed members:
  - g_rate //the grooming rate
  - P_groom_balance // probability to select the groomed person according
to groom balance
  - num_posts // the total number of posts

for (i=0; i<num_posts; i++):
  //select a speaker (post maker) out of group members
  if rand() > P_nys:
    Select a not-yet-spoken member
  else: //select a speaker out of the yet-spoken members
       if rand() > P_groom_sum:
         Select the speaker with the max amount of grooms received
       else: //select a speak according to their grooming status
         if rand() > P_groomed:
              Randomly select one from the groomed group members
         else:
              select one from the ungroomed members
  // Decide if a post contains a groom
  if rand() > g_rate:
       Attach a groom to the post
       //Decide whom is groomed if the post contains a groom
       if rand() > P_groom_balance:
         Groom the member owning the max amount of groom balance
       else:
          Groom the member made the max amount of posts
```

Figure 5.3 Group interaction simulator

Roles detector detects individuals' role in each group. Referring to the aforementioned studies (cf. SNA in Section 4.1.4.2), this thesis retrieves totally six roles: *leader*, *disseminator*, *responder*, *broker*, *lurker* and *peripheral*. The mapping criteria to their corresponding roles can refer to Table 5.8. With regard to the workflow to detect these pre-defined roles, Figure 5.4 depicts it.

```
parameters:
  - num_groups //the number of groups
  - low_bound
  - high_bound
for (i=0; i<num_groups; i++): // each group
  draw the i_th group's social network
  for (j=0; j<size(i); j++): // each member in this group
    calculate the j_th student's SN metrics
  for each SN metric k:
    calculate the min (min_i_k) and max (max_i_k) in this group
  for (j=0; j<size(i); j++):
    for each SN metric k:
       if k_j = 0: // the j_th student's k_th SN metric
             k_j = 'Null'
       else if min_i_k<=k_j<(max_i_k-min_i_k)*low_bound+min_i_k:
             k_j = 'Low'
       else if (max_i_k-min_i_k)*low_bound+min_i_k<k_j<=(max_i_k-
min_i_k)*high_bound+min_i_k:
             k_j = 'Medium'
       else:
           k_j = 'High'
    map the j_th student's role according to the given criteria
```

Figure 5.4 Workflow of roles detection

Machine learning component takes group cohesion and group roles as input data to train a classifier that is able to predict the group outcome of any new group. Specifically, it sums up the number of students for each role in each group and regards the group cohesion as the target. The median of group cohesion was taken as a division to separate successful groups and weak groups. A snippet of the data set can be seen as Table 5.3. Note that the target label ‘1’ indicates a successful group while ‘0’ tells us a weak group.

Table 5.3 Snippet of data set

	#disseminator	#responder	#broker	#leader	#lurker	#peripheral	#unknown	target
Group#1	2	0	1	1	2	2	0	0
Group#2	0	4	0	2	2	2	0	0
Group#3	0	2	1	1	1	5	0	1
…	…	…	…	…	…	…	…	…

Classifier is a product of the machine learning component (i.e. a decision tree, cf. Figure 5.5). It tells whether the newly formed groups are successful or not. For example, the decision tree in Figure 5.5 indicates three group composition rules: 1) if the number of lurkers is not more than 2.5, the groups could be successful; 2) if the number of lurkers is more than 2.5 and the number of leaders is more than 1.5, the groups would then be good too; 3) if the number of lurkers is more than 2.5 and the number of leaders is not more than 1.5, we then would get unsuccessful groups. We desire successful groups rather than low-performing ones, and the more, the better. Thus, the total amount of predicated successful groups is a measure to evaluate the quality of the resulting group formations.

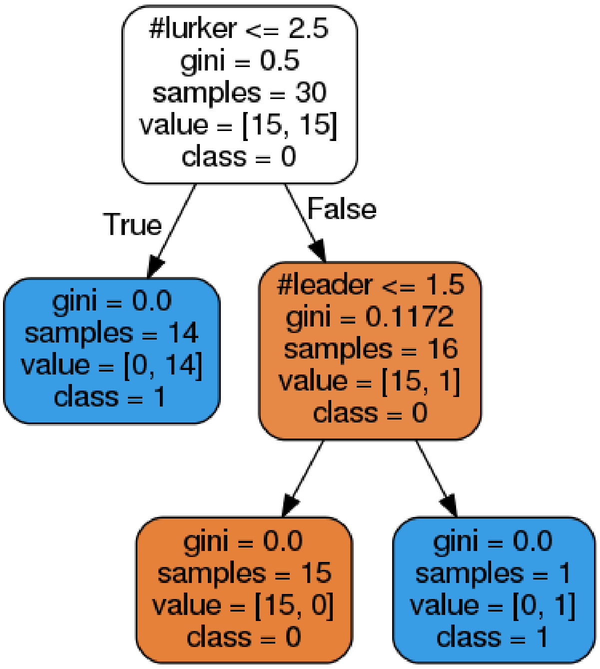


Figure 5.5 Decision tree sample

```
parameters:
  - nIter // the number of optimization iterations
  - nSwarm // the number of PSO particles
  - S_leaving // a set of students who want to leave for new groups

randomly create nSwarm group formations with students in S_leaving
for(i=0; i<nIter; i++):
  apply the classifier to evaluate each group formation's quality and
employ the discrete-PSO algorithm to improve the group formations
return the best formation over the swarms and over the iterations.
```

Figure 5.6 Workflow of regrouping component

Regrouping component creates new group formations for ready-to-leave students and iteratively optimizes them with respect to the quality of such evaluated by the classifier. A discrete-PSO algorithm (cf. Section 3.2.2.1) is applied to perform optimization.

5.2.1.3 Parameters

Group formation parameters define how big the group formation is. In current MOOCs, thousands of students are normally enrolled in each. Besides, the group size also, to some extent, determines the complexity. The bigger the group is, the more pair compatibilities have to be calculated. In a pedagogical respect, group size certainly has to be set up according to the difficulty level of tasks. The easy tasks need fewer group members than the difficult and complex tasks. Another important factor to be taken into account in open learning environments (e.g. MOOCs) is students' drop-out. The smaller groups might be less likely to achieve their group goals in the end because the labor force decreases with the high drop-out rate.

Table 5.4 Group formation parameters

Number of students	10,000
Group size	10

Drop-out parameters simulate how many students drop out and how often they are supposed to be removed in the simulation experiment. Note that drop-out students are, by definition, the most inactive students in groups. Suppose a group task follows the pace of most MOOCs' weekly releasing mode, a weekly drop-out rate needs to be defined. Referring to the relevant research work on MOOC dropout (Balakrishnan, 2013; Kloft, Stiehler, Zheng, & Pinkward, 2014; MOOCs@Edinburgh-Group, 2013), the vast drop out of MOOC students normally occurs in the first week (approx. 50%) followed by a comparatively smaller yet stable decline rate in the following few weeks. Based on such empirical evidence, two points are inspired. First, the first week would undoubtedly not be a desirable fit to group work because of that foreseeable group instability. Second, from the second week onwards, we can estimate that the weekly drop-out rate could range from 0 to 50%. Still, we do not know how much exactly that weekly drop-out rate is. In practice, the answer should vary depending on the courses. With that in mind, there is no harm to set the weekly drop-out rate to a random float number ranging from 0 to 50%. Drop-out actually could happen at every moment, but for convenience, the drop-out students in this simulation system are removed (compel those students out of the simulation system) every day. The due point of removing students is at the end of each day. Since the simulation is not time-based but event-based, that is, the group work progresses by the number of interactions rather than the timeline, the exact value of this parameter is the average number of events per day. In other words, after such number of events, a day is presumably over.

Table 5.5 Drop-out parameters

Weekly drop-out rate	A random in-between (0, 0.5)
Daily drop-out rate	Weekly drop-out rate/7
Due of removing drop-out students	10

Regrouping parameters define how many students from four different categories would leave the current groups for the newly composed ones. Categorizing the students is based on their participation and group performance. Active and inactive participation together with successful and unsuccessful group performance compose the four different categories in this set of parameters. Note that this thesis does not recompose all participants into new groups. Hence, it is necessary to select students who could probably leave for new groups with these parameters. The baseline of defining the leaving rates is threefold. First, active students should be more likely to leave for new groups than their inactive counterparts. Second, unsuccessful students should desire more chances to be successful via joining new groups. The aforementioned two points imply that the active but unsuccessful groups might more likely leave for new groups than the others. In contrast, the most conservative but 'most clever' could be the inactive but successful students. They, in fact, take full advantage of the group work without any substantial effort. They could continuously enjoy the benefits as a free rider so as not to leave the current groups soon before any negative outcomes occur. Based on the upper assumptions, we can simply randomize the leaving rates of those four categories of students (ranging from 0 to 1), but they should, in the meantime, follow the aforementioned three laws (see the detailed settings in Table 5.6).

Table 5.6 Regrouping parameters. Note that the active but unsuccessful students' leaving rate is greater than the others and the inactive but successful students' leaving rate is the lowest

Active but unsuccessful students' leaving rate	A random in-between (0, 1)
Active and successful students' leaving rate	A random between the active but unsuccessful students' leaving rate and the inactive but successful students' leaving rate
Inactive but successful students' leaving rate	A random between 0 and the active but unsuccessful students' leaving rate)
Inactive and unsuccessful students' leaving rate	A random between the active but unsuccessful students' leaving rate and the inactive but successful students' leaving rate

Group interaction parameters are listed in Table 5.7. First of all, the number of posts sets up the total volume of posts that each group can make in the group discussion forum. A normal distribution of such is assumed over all groups. A standard deviation of such data is then needed as well. Second, the grooming rate signifies the number of grooming posts. A grooming post is a post that includes social grooming (e.g. to reply to a posting or to cite sayings from someone). In the same fashion as the number of posts, a standard deviation of this parameter is also needed because different groups vary in the quantity of grooming posts. According to Nygren's simulation model (Nygren, 2012), there are two ways to select a speaker to make a post in general. First, not-yet-spoken group members can be selected; second, already-spoken group members of course can make a new post once again. Selecting either out of the both is probabilistic. Moving our focus onto already-spoken group members, groomed and ungroomed members out of them should have a different probability to speak. As Nygren observed, the likelihood to make a new post is not merely associated with the status of being groomed but also in conjunction with the number of grooms they

Table 5.7 Group interaction parameters

Number of posts	70
Standard deviation of the number of posts over all groups	20
Grooming rate	0.6
Standard deviation of the grooming rate over all groups	0.05
Probability to select a not-yet-spoken member as a new speaker	0.15
Probability to select a speaker according to the grooming status	0.33
Probability to select a speaker according to the sum of grooms received already	0.67
Probability to select a speaker out of ungroomed members	0.27
Probability to select a speaker out of groomed members	0.72
Max number of initial interacting agents	5
Due of deactivating the over-spoken member	10
Roles stereotyping rate	0.5

already received. The more historical grooms unveil more potential to speak. The maximal number of initial interacting agents controls the spread of interaction over the whole group. The group interaction merely confines to very few of those group members if no control over the number of initial interacting agents exists. The due of deactivating an over-spoken person is used to avoid the case that someone always seizes the loudspeaker so as to dismiss others' opportunities to speak. At last, the roles stereotyping rate is a probability to boost such active roles as leader and disseminator to make a new post. This is only applied to the tasks after the first one. Nygren did not observe the participants' behaviors in the subsequent tasks. The modeling is reported to be of Monte-Carlo type. If we run this modeling for the subsequent tasks the same as the first one, this would, as a result, lead us to randomness. And this actually does not comply with our assumption that the group roles play an important part in group interaction. To avoid this, we applied an additional policy for the tasks after the first task. The policy is that leaders and disseminators have some privileges to make a new post (determined by this stereotyping rate). As shown in Table 5.7, except for the roles stereotyping rate, we borrowed almost all of the parameter settings from Nygren's work (Nygren, 2012).

Group members' roles parameters give criteria to detect individuals' roles in learning groups. In total, 6 roles are defined in this thesis, namely, *leader*, *disseminator*, *responder*, *broker*, *lurker* and *peripheral*. As inspired from the previous studies (Capuano et al., 2014; Marcos-García et al., 2015), leveraging SNA makes it possible to infer implied roles in social networks as long as mapping between the inspected roles and the social network metrics is empirically established. Leaders normally remain very close with other members in groups because they not only need to do management jobs, but also coordinate group members' activities in part or whole. They are thus supposed to stand at the center of the social networks. More specifically, their closeness centrality (a metric of SN) in the social networks should be high and so should the overall degree. Disseminators' role is to ask questions or make new posts. Besides, their posts normally attract much attention, that is, they always start good (might be on-topic) discussions. Thereby, their closeness centrality and in-degree are not supposed to be low. On the contrary, responders take the responsibility of answering questions and actively joining the newly created discussions. Their closeness centrality and out-degree can thus be thought of as higher than the low level. Brokers act as a key person to transmit messages from the one relatively close cohort to another close one. They essentially make critical intra-group connections among different cohorts, which in general can bring complementary points of view to one another. As such, their betweenness centrality in the social networks is rather high. The difference between the remaining two negative roles, the peripherals and lurkers, is whether they make contributions to the groups. Lurkers never actively join the discussions, but perhaps they sometime view group activities whereas the peripherals make at least some contributions during the group process, but still comparatively fewer than those with active roles. The overall mapping criteria are shown in Table 5.8.

Table 5.8 Criteria to map group roles and SN metrics. Note that the boundary of each level is Low: [0, 0.2], Medium: (0.2, 0.7], High: (0.7, 1].

	Degree	In-degree	Out-degree	Closeness-centrality	Betweenness-centrality
Leader	High	Medium/High	Medium/High	High	
Disseminator	Medium/High	Medium/High		Medium/High	
Responder	Medium/High		Medium/High	Medium/High	
Broker					Medium/High
Lurker	Null	Null	Null	Null	Null
Peripheral	Low	Low/Null	Low/Null	Low	Low/Null

PSO parameters relate to the setting-up of the discrete-PSO algorithm that is used to optimize the newly composed group formations iteratively. The swarm size indicates how many particles would be taken into computation. In principle, more particles join the computation. It would then get more optimal options. However, it then has to pay the price of time cost. The number of iterations is how many iterations you want the algorithm to run. The perfect solution is when the whole optimization process converges. However, for the large scale problems, especially those running on real-time systems, it has to find a trade-off between the time cost and the optimization quality. The inertia weight and acceleration constants are specific to the inner working mechanism of the PSO algorithm. They generally control movement of each particle towards the 'best' positions altogether (cf. Section 3.2.2.1).

Table 5.9 PSO parameters

Swarm size	30
Number of interactions	200
Inertia weight	0.1
Acceleration constants	C1: 0.1 C2: 0.5

5.2.2 Observables

To examine the proposed group re-composition's impact via the simulator, two indicators were selected to be observed, namely group cohesion and drop-out. Note that we could not use learning performance as an indicator in our simulation (as no learning was modeled). Alternatively, group cohesion was selected as a substitute for the learning performance for the sake of their positive relationship to each other. Dating back to the 1990s, Evans and Dion found a positive relationship by the use of meta-analysis (Evans & Dion, 1991).

Group cohesion, as a structural measure of social network, in this work is defined as the number of the actual inter-ties among group members divided by the total number of possible ties between any pair of members (S. Wise, 2014). By definition, group cohesion directly reflects inter-person ties in groups. The higher the group cohesion is, the stronger ties the groups have. Such strong ties, as inter-communication pipelines, undoubtedly address the salient problem of information-sharing and knowledge-sharing. Group members can thus know each other better and fulfil their common goals faster as a result (Hansen, 1999; Lechner, Frankenberger, & Floyd, 2010). Another interesting work from Tsai and Gohshal (Tsai & Ghoshal, 1998) points out that group cohesion, as a structural measure, does not only increase inter-personal trust from the social perspective, but also plays an important part in both forming and sharing common goals and values among group members. Overall, group cohesion mirrors group performance. It was thereby chosen as an indicator of learning performance in this simulation work.

Drop-out is another important indicator that reflects students' engagement. In current MOOCs, students' engagement is reported to be associated with course content and personal motivations. In group work, what factors can explain such have never been studied thus far. The observation, on the one hand, could hopefully give us some unseen hints to address the high drop-out problem. On the other hand, it could reveal the proposed group re-composition's impact.

5.2.3 Experimental procedure

All relevant parameters were set as their default value described in this thesis. As shown in Figure 5.7, 10,000 students' (1,000 groups) group interactions were simulated in the first task. In the meantime, the dropout students were regularly removed. After their interaction, the students who desired a new group were selected.

They were then composed into new groups using the proposed approach. In order to highlight the results in a comparable fashion, the same amount of students were copied and composed into random groups. The former is named algorithmic condition and the latter is named random condition. In the real world, this is certainly not practically feasible. In simulation, it is, however, fairly easy and allows us to simulate what happens to the same students when re-grouped either algorithmically or randomly. All those new groups' interactions were next simulated in Task 2 and the dropout students were removed again. When the second task was over, the dropout students and cohesive groups were counted for both the algorithmic condition and the random condition. Note that a cohesive group is a group with a group cohesion that is higher than the median cohesion over all groups in both conditions. We ran the whole process 10 times to avoid any biased result probably generated by chance.

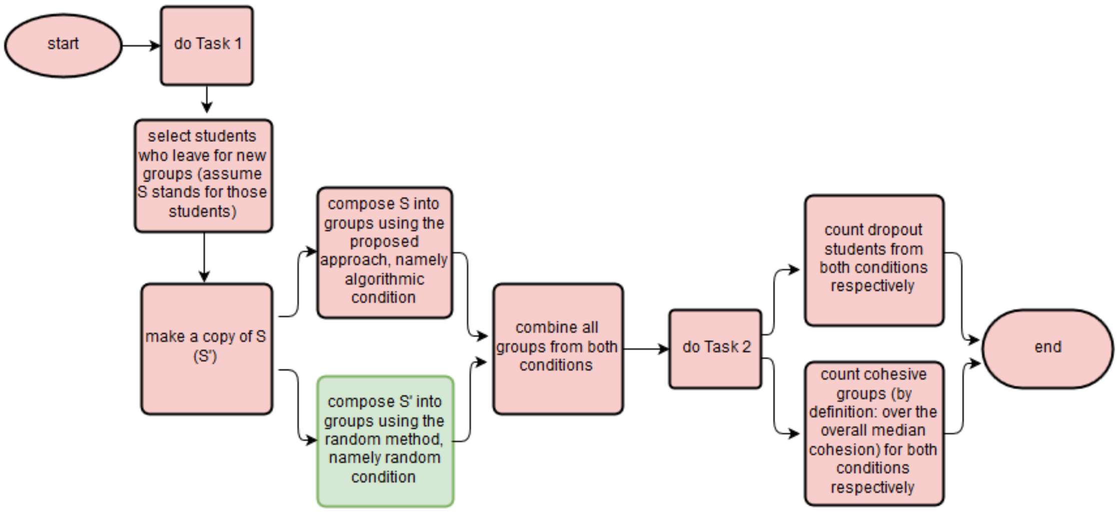


Figure 5.7 Experimental design

5.2.4 Experimental results

Table 5.10 presents the group cohesion and drop-out. Recall that the whole simulation was repeated 10 times intending to avoid by-chance results. As shown in the table, the ratio of cohesive learning groups in the algorithmic condition is larger than in the random condition (algorithmic: 0.466 vs random: 0.421). Note that the ratio of cohesive groups is the proposition of the cohesive groups over the total number of the newly composed groups. Since both algorithmic and random conditions have same amount of the newly composed groups, higher ratio simply means more cohesive groups. A t-test was performed and the result indicates that the algorithmic condition produced significantly more cohesive groups than the random condition (p-value: 0.035).

Regarding drop-out, more drop-out students came from the random groups than from the algorithmic ones (algorithmic: 0.468 vs random: 0.530). Note that this dropout ratio does not mean 46.8% students out of the algorithmic groups dropped out while 53.0% students out of the random groups dropped out. Rather, it tells that, out of every 100 dropout students, 47 students came from the algorithmic condition while 53 students came from the random condition. Likewise, a conducted t-test indicates a significant difference between both conditions (p-value: 0.002).

Table 5.10: Ratio of cohesive groups and drop-out. A total of 10,000 students and repeatedly run 10 times

	Ratio of cohesive groups		Raito of dropout students	
	Algorithmic	Random	Algorithmic	Random
Runtime #1	0.447	0.335	0.428	0.571
Runtime #2	0.482	0.476	0.453	0.546
Runtime #3	0.420	0.411	0.445	0.554
Runtime #4	0.504	0.488	0.537	0.462
Runtime #5	0.440	0.36	0.503	0.496
Runtime #6	0.503	0.468	0.446	0.553
Runtime #7	0.486	0.464	0.465	0.534
Runtime #8	0.447	0.419	0.516	0.483
Runtime #9	0.480	0.360	0.416	0.583
Runtime #10	0.456	0.431	0.416	0.522
Ave.	0.466	0.421	0.468	0.530
SD	0.026	0.051	0.037	0.037
p-value	0.035		0.002	

Although learning performance was not observable in this simulation experiment, it is not difficult to infer the positive impact on learning performance based on the reported positive ties between group cohesion and learning performance. Since we also saw a lower drop-out rate in the algorithmic groups, the impact on decreasing the drop-out rate appears positive too.

5.3 Discussion

5.3.1 Participation

Massive enrolment (thousands of participants) is one of the typical (and defining) MOOC features. Testing re-grouping methods on a data set of 10,000 students as we did is thus obviously necessary. However, one could ask whether the approach also works with a few thousand participants or with even smaller courses, such as an on-campus moodle course with a few hundred students. In an attempt to answer such a question, another two simulation experiments were run with three thousand students and one hundred students respectively.

When the participation is set to three thousand students, the observation of drop out and group cohesion reveals no difference to the case of ten thousand students. Similarly, more cohesive groups came from the algorithmic condition than the random one (average: 0.474 vs 0.428, cf. Table 5.11). Students were more likely to drop out in the random condition (0.475 vs 0.524). A statistical test again confirmed significant differences (ratio of cohesive groups: p-value = 0.011, dropout: p-value= 0.001).

In the case of one hundred students, the drop-out rate found in both conditions is almost same (average: 0.501 vs 0.498, p-value: 0.959, cf. Table 5.12). The reason for the deficits, in this case, has its roots in the very limited number of groups to be composed. The number of newly composed groups should be fewer than the total 10 groups (group size is set to 10). In such a small possibility scope, there is no need to challenge the algorithm's capability. In other words, the algorithmic method could perform no better than a random method in such a small case.

Varying the course size from one hundred to ten thousand over these three experiments, the observations tell us two points. First, the scale of participation does matter for the simulation results. Second, the proposed data-driven approach seems to make a positive effect beyond a certain level of participation. At least, for a hundred students, it does not reveal any of its superiorities.

Table 5.11 Ratio of cohesive groups and drop-out. A total of 3,000 students and repeatedly run 10 times

	Ratio of cohesive groups		Raito of dropout students	
	Algorithmic	Random	Algorithmic	Random
Runtime #1	0.451	0.354	0.441	0.558
Runtime #2	0.488	0.466	0.458	0.541
Runtime #3	0.494	0.494	0.437	0.562
Runtime #4	0.508	0.428	0.479	0.520
Runtime #5	0.484	0.473	0.490	0.509
Runtime #6	0.446	0.446	0.531	0.468
Runtime #7	0.512	0.410	0.481	0.518
Runtime #8	0.470	0.400	0.459	0.540
Runtime #9	0.457	0.414	0.477	0.522
Runtime #10	0.428	0.397	0.500	0.500
Ave.	0.474	0.428	0.475	0.524
SD	0.026	0.039	0.026	0.026
p-value	0.011		0.001	

Table 5.12 Ratio of cohesive groups and drop-out. A total of 100 students and repeatedly run 10 times

	Ratio of cohesive groups		Raito of dropout students	
	Algorithmic	Random	Algorithmic	Random
Runtime #1	1.000	0.000	0.333	0.666
Runtime #2	0.333	0.333	0.500	0.500
Runtime #3	0.33	0.333	0.666	0.333
Runtime #4	0.333	0.666	0.666	0.333
Runtime #5	0.500	0.500	0.388	0.611
Runtime #6	0.600	0.400	0.541	0.458
Runtime #7	0.600	0.400	0.583	0.416
Runtime #8	0.666	0.333	0.500	0.500
Runtime #9	0.666	0.333	0.500	0.500
Runtime #10	0.571	0.285	0.333	0.666
Ave.	0.560	0.358	0.501	0.498
SD	0.194	0.159	0.114	0.114
p-value	0.027		0.959	

5.3.2 Dropout

The evidence of fewer drop-out students coming from the algorithmic condition than the random condition is interesting to interpret. First, the proposed group re-composition approach is able to learn the best role matching patterns in groups while the random approach cannot. For instance, a group with many disseminators but without responders might not achieve success in the sense that few questions would be answered though many would be asked. If such an assumption is true, the proposed data-driven approach can then quickly gain this knowledge from the data too. It would then try to avoid such when it composed new groups. But the random approach never notices this and could thus run into a bad group composition over and over again. Those randomly, often badly, composed groups could bring about much more inactive students so that more drop-out arises in the end. Recall the assumption that drop-out students are the most inactive students.

Second, the proposed approach can increase group cohesion, which implies that the ties among group members are strengthened as well. Such strong social connections could be an incentive to stop students from leaving. If it is factually true, this could inspire us to have another look at the current MOOCs. Many of the current MOOCs integrate a public discussion forum as a platform that basically scaffolds students' discussions on course-related issues. However, very few actively participate in such discussions (5-10%) (Rosé & Siemens, 2014). This, from another side, shows us the great scarcity of social connections among all participants, which could be an important factor to explain the factual high drop-out rate if contrasting it with the findings in this thesis.

Even though the improved social ties in small discussion groups could mitigate the high drop-out problem, the integration of small group discussion into current MOOCs also has to be done very carefully. First of all, how to deal with the public discussion forum needs to be discussed. As many MOOC participants may know, the public discussion forum plays an important role in scaling up knowledge sharing. In this respect, small group discussion, of course, is a loser. Therefore, a simple removal of the public discussion forum seems to be unreasonable. An alternative is to regard small group discussion as a complement to the public discussion forum. Students could be encouraged to start with small group discussion. If answers to some of the discussion topics do not satisfy themselves, they can choose to share them with the public in order to collect much more opinions via crowdsourcing. Second, many MOOCs do not assign specific group tasks to small groups. Without a group goal, it is still questionable whether those groups can functionally maintain themselves as a whole. Artificial agents of Q&A could be applied to such a case, aiming to activate group discussions if necessary. However, students' interactive enthusiasm has to be investigated as well.

5.3.3 Time cost

It is also interesting to see how much time the proposed approach costs. The time cost depends on how many students would be willing to leave for new groups. More students would make the problem scope bigger and thus needs more time in the end. As mentioned in Section 5.2.1.3, the regrouping parameters determine how many students would leave for new groups. Thus, this thesis experimented with setting those parameters to three different scales (i.e. 0.3, 0.6 and 0.99). If those parameters are 0.3, that means 30% of students would leave for new groups. If those parameters are set to 0.99, that means roughly all participants would leave for new groups, which could be a good fit to scenarios that forcefully compel all students to new groups. The enrolment was still set to 10,000 students as well in the experiment. With each setting, the experiment repeatedly ran 5 times.

Table 5.13 Time cost of the regrouping approach. A total of 10,000 students were simulated and it repeatedly ran 5 times.

	Time cost (in seconds, leaving rates: 0.3)	Time cost (in seconds, leaving rates: 0.6)	Time cost (in seconds, leaving rates: 0.99)
Runtime #1	1945	6674	20943
Runtime #2	2295	7811	22514
Runtime #3	2327	8279	25208
Runtime #4	2745	10160	25398
Runtime #5	2818	10317	30903
Ave.	2426	8648	24993
SD	320	1400	3396

As shown in Table 5.13, it takes a couple of hours (roughly 6.8 hours on average) to recompose the 10,000 course participants into new groups. Even with 30% students, it needs more than half an hour. Certainly, these experimental results are constrained by my personal computer (an Ubuntu virtual machine (2 CPUs and 4098M memory, the host computer with Intel(R) Core(TM) i7-4600U 2.10GHZ CPU and 8G memory)). But this might suggest to us that instant re-composition of those thousands of students could not be possible using the proposed approach. For larger-scale problems (i.e. more than 10,000 participants), we are able to estimate its time cost, because the time cost increases linearly as the scale grows (cf. Figure 5.8). To reduce the time cost, if necessary in some cases (e.g. with a few hours between two consequent group tasks), some of the recent cluster computing systems (e.g. Apache Spark[29]) could be helpful.

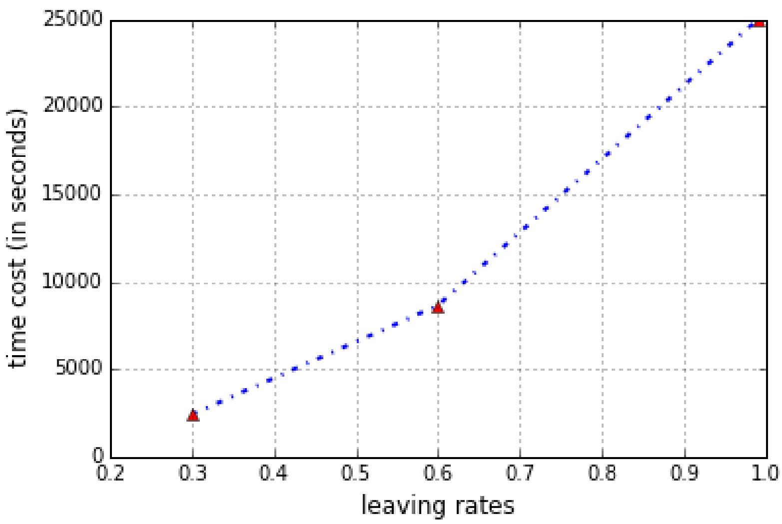


Figure 5.8 Average time cost of different scales

5.3.4 Classifier

The approach relies on the classifier in the sense that it can tell us which group composition is successful and which is not. The performance of the classifier thereby plays a vital role in the whole process. To examine its accuracy, a 5-fold cross-validation was performed. As shown in Table 5.14, the classifier's accuracy comes up to 0.90 on average. Compared to the baseline that always predicts the majority class, the selected decision tree classifier evidently wins by a large margin. Nevertheless, the decision tree classifier has its own disadvantage, such as the overfitting problem. The overfitting problem could make the classifier biased to the training data and thus reduce its accuracy for the test data that it never saw. To mitigate such, constraining the depth of the trained decision tree could avoid overfitting to the training data to some extent. The concrete depth value could depend on the scale of data, however, this must be experimented in the future.

[29] http://spark.apache.org/

Table 5.14 Prediction accuracy

Runtimes	Decision Tree		Baseline
	Accuracy	SD	Accuracy
#1	0.90	0.01	0.60
#2	0.90	0.01	0.58
#3	0.88	0.02	0.61
#4	0.91	0.02	0.58
#5	0.91	0.02	0.58
#6	0.91	0.01	0.54
#7	0.90	0.02	0.57
#8	0.86	0.01	0.6
#9	0.90	0.01	0.62
#10	0.90	0.02	0.58
Ave.	0.90	0.02	0.59

5.3.5 Internal validity

The internal validity is often used to measure the correctness of implementing the simulation model. Simulation, more specifically here, a computer simulation, is programed in computer instructions, which aims to implement a set of assumptions and fulfil their effect as a result. As mentioned in the beginning of this chapter, rational associations between the inputs and outputs are hardly found in problems that simulation tries to address. It is thereby difficult to validate the consequences of such computer implementation (Axelrod, 1997). In this thesis, the heart of the simulation is the group interaction data generated by Nygren’s online group discussion model. It is necessary to take a close look at such data and intuitively judge its validity. Around this, I intend to showcase the data from three different aspects: group cohesion, drop-out and group roles.

First, group cohesion over all groups is expected to follow normal distribution. Many groups center around a medium level of interaction. As shown in Figure 5.9, the median almost evenly splits the data points and the middle half of those are closely around the median, which obviously satisfies our assumptions. In addition, one may notice another fact that the maximal group cohesion is far away from the full scale (i.e. 1.0), which implies that the group members are not completely connected. This could cast doubt onto the idea that group interaction was fully exploited. Two factors could probably explain such a phenomenon. First of all, leadership that is normally found in strongly tied groups could influence the structure of group interaction (cf. Figure 5.10). These centralized networks with one or even more leaders at the center convey knowledge sharing through a central hub instead of a peer-to-peer fashion. Such a networking structure should be more similar to learning groups and project teams in real world than a complete-graph structure. Second, the parameter (i.e. number of posts) somehow limits the connections between group members. If we tune up this parameter, more new peer-to-peer connections could be produced in some of those groups. Nevertheless, some of those ‘well-done’ groups (with a very cohesive interaction circle) could be immune to such an adjustment.

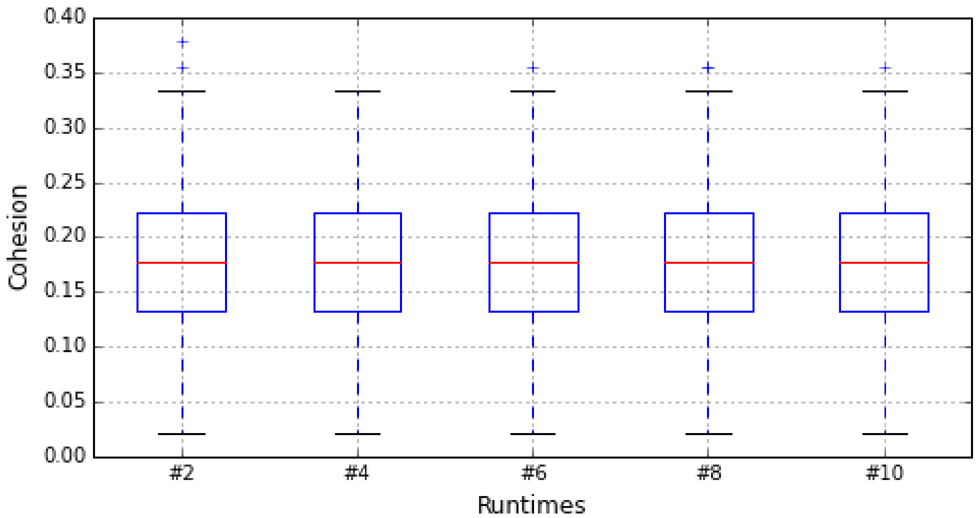


Figure 5.9 Distribution of cohesion. The data from the 2nd, 4th, 6th, 8th and 10th run times were collected and the data is from the initially composed groups

Second, it was assumed that the most inactive students would drop out from their groups. The typical structure of intra-group sociograms is shown in Figure 5.10. The number inside each node can be considered as the students’ unique id. Edges represent the connections between group members. The green node is used to separate the drop-out students from the remaining students in red colour. As we can see, the drop-out students are always the ones with few or no contributions to the group interaction.

Another thing of interest is that some groups are strongly connected (e.g. group #1, #6 and #8) whereas some others are not (e.g. group #2 and #5), which of course again confirms the assumption of various levels of group interaction.

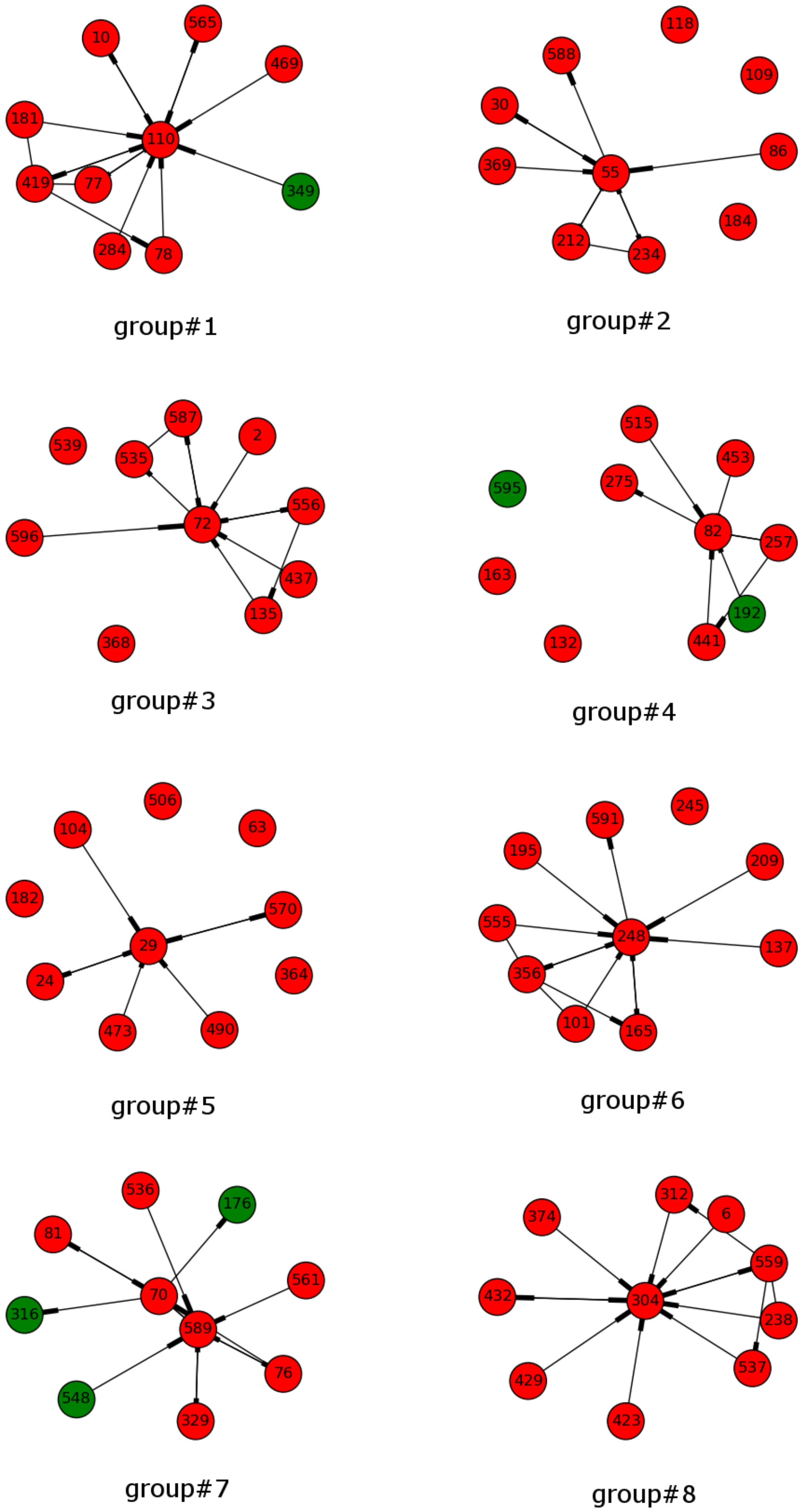


Figure 5.10 Sociogram examples

Lastly, the roles defined in this thesis were all successfully detected from the group interaction data. More than that, Figure 5.11 tells us that lurkers and the peripherals take the vast majority, which complies with the current reports on MOOC. Aside from these, almost 10 percent of leaders were identified among groups, which literally means that each group has a leader on average (of course, weakly connected groups should not, and strongly tied groups may have more than one leader). This, on the one hand, abides by the reality as discussed beforehand. On the other hand, it could oversimplify the reality. Other sorts of networking structures could probably exist in the real world too. At last, the detected disseminators account for a very small proposition (approx. 2%), which does not mean a very small amount of questions are asked. Instead, it indicates that few interesting questions that attract much attention from group members were given.

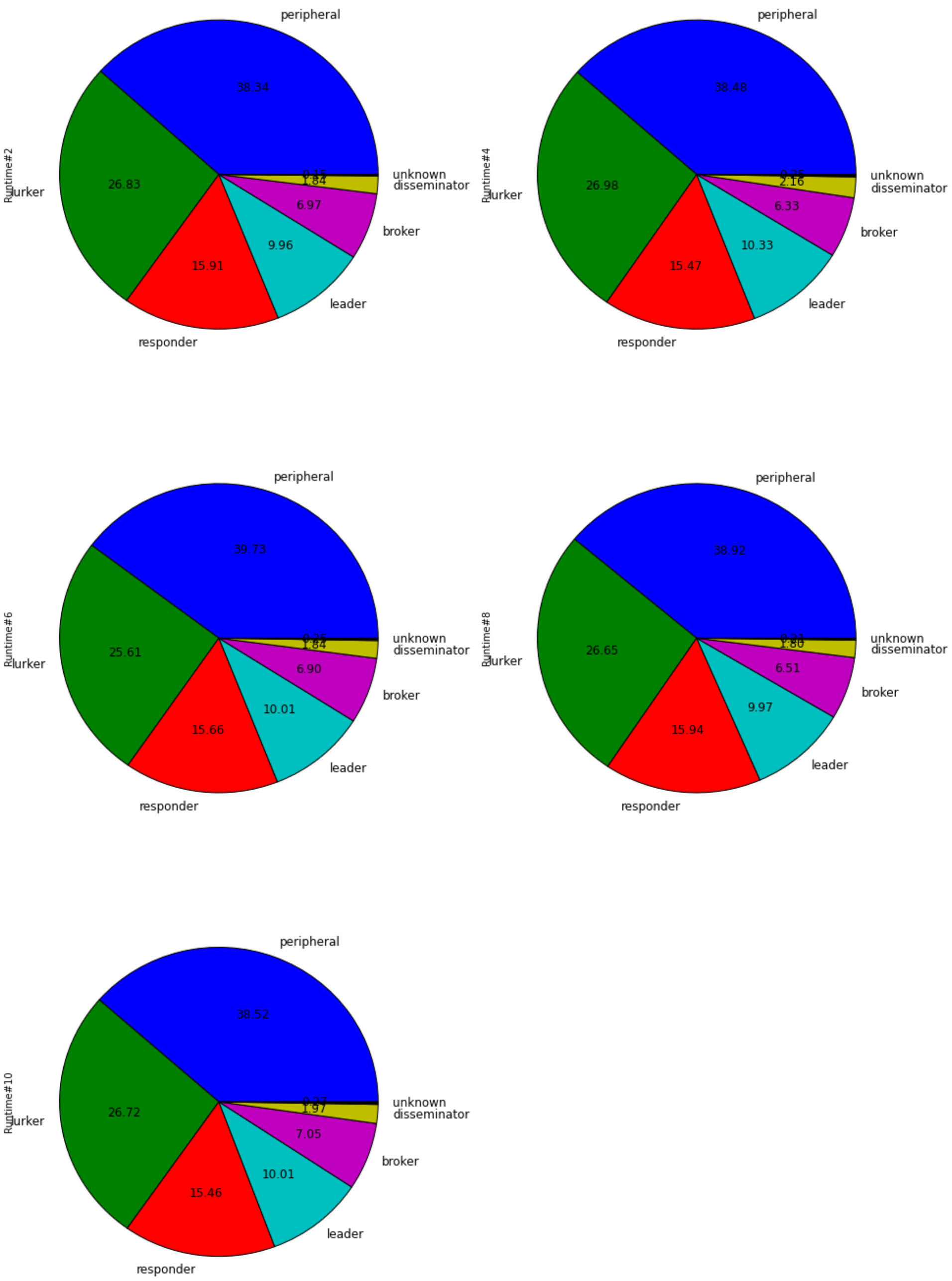


Figure 5.11 Roles distribution. The data from the 2nd, 4th, 6th, 8th and 10th run times were collected and the data is from the initially composed groups

5.4 Chapter summary

This chapter details a simulation experiment that is used to evaluate the proposed group re-composition approach. The key part of this experiment is a simulation system that is capable of implementing the proposed group re-composition approach. The input of the simulation system is a population of students. The system is then able to compose those students into initial small learning groups, simulate group interaction, detect each individual student's role, induce the group composition rules and re-compose students into new groups. Since the presumable application scenario is a MOOC course, some of the typical features in MOOCs (e.g. dropout) are also simulated.

In the simulation experiment, two conditions, namely a random condition and an algorithmic condition, were designed. In the random condition, the learning groups in the simulation system were re-composed randomly. In the algorithmic condition, the proposed group re-composition approach was adopted to re-compose learning groups. The experimental results indicate that the algorithmic condition yields more cohesive groups and a lower dropout rate than the random condition.

6 Conclusions and Outlook

6.1 Summary

This thesis seeks to shed some light on the creation of small learning groups in recent MOOCs. Recent MOOCs feature massive numbers of students. Although many group composition approaches have been previously suggested, few of them have been tested on a big data set. This thesis re-examines some of those approaches from the perspectives of time cost and grouping quality with a MOOC data set. Based on the experimental results, some suggestions on how to select a suitable group composition approach have been given. (cf. Chapter 3).

A MOOC course, normally, has a large number of drop-out students throughout its lifecycle. As a result, many learning groups would be incomplete in size. Thus, making small learning groups in a one-off fashion does not fit to such open learning environments as MOOCs any longer. In other words, maintaining groups become challenging in the end. To mitigate such a problem, a dynamic group re-composition approach is proposed in this thesis. This approach is not simply applying the existing group composition methods multiple times. Rather, it automatically gains insights from group interaction data together with group outcome data and reapplies such insights to make new groups one time by another. This approach is of data-driven nature and makes full use of data intelligence during the course of group process. It is different than the existing grouping methods because of its removal of reliance on the pre-defined group composition criteria.

Regarding the implementation of the group re-composition approach, this thesis illustrates design and development of a group tool in recent MOOC platforms. The group tool does not merely fulfil the goal of group re-composition, but also offers group learning facilities. Students can discuss and collaboratively write their solutions to the assigned group tasks in small groups (cf. Section 4.2 and 4.3).

A computer simulation experiment was conducted to evaluate the group re-composition approach. In such a simulation environment, group interaction in MOOC groups was structurally simulated. Besides, the students' drop-out patterns are also parameterized. Through a close look at the drop-out and group cohesion, the experimental results suggest that the proposed group re-composition approach can reduce the drop-out rate and bring more cohesive learning groups than a random condition.

When looking back on the research questions defined in Section 1.2, the findings of this thesis can be concluded more specifically as follows:

- **RQ1:** What methods are employed to compose small learning groups, and which out of those group composition methods could be suitable for large-scale learning settings and what about their efficiency?

Generally, random groups, self-selection groups and algorithmically-composed groups have been widely made in a variety of learning contexts over decades. Random groups and self-selection groups are comparatively easy to be made. However, their difficulties to fulfil specific grouping criteria (e.g. ability heterogeneous groups) could make them exclusive from some educational practitioners' consideration (cf. Section 2.3.1). In contrast, computer algorithms can deal much better with various grouping criteria (homogenous, heterogeneous and even constraint-based group criteria). Thus far, a wide range of algorithms have been proposed to solve the group composition problem (cf. Section 2.3.2). Owing to the growing number of students in recent large-scale learning platforms (normally up to thousands of students), many of the existing grouping algorithms have not been tested in such a large scale, this thesis thus selected four typical group algorithm (namely, a discrete-PSO algorithm, a genetic algorithm, an adapted k-means algorithm and an ant colony optimization algorithm) and re-examined their efficiency over both time cost and grouping quality with a MOOC data set. The results suggest that there is a trade-off between time cost and grouping quality. The ant colony optimization algorithm won the other three in quality of the grouping solution but costed much more time (approx. 25 times as much as the genetic algorithm, 8302 vs 340 seconds, see details in Section 3.3.2). Nevertheless, if the defined criteria are far beyond the scope of heterogeneous and homogeneous types, for instance, tree-like structured criteria (cf. Figure 5.5), the ant colony optimization algorithm and the adapted k-means might not be applicable because of their heavy reliance on calculating Euclidian distance between two data points (students) when they conduct local search. The genetic algorithm and the discrete-PSO are immune to such because they focus on the index of each data point rather than the peer distance. Thus, in such situations, the recommendation is the discrete-PSO because of its superiority in grouping quality over the genetic algorithm (see details in Section 3.3.2).

- **RQ2:** How can one apply a group composition method to a MOOC course and what is the impact of group composition on dropout and learning performance?

As mentioned in Section 3.4, one can simply replicate the same approach as I did in that MOOC experiment. First, send out a pre-course survey (see Appendix Ⅴ) to the course participants. Second, collect the survey data. Third, use a group composition algorithm to compose learning groups. Lastly, inform students of their group members' contact. The benefit of this method is no need of a third-party group tool. Students can leverage the existing social media channels and email to start group discussion. The downside is the lack of learning-related moderation. Students stay in their own social media groups. They may make many off-topic discussions or no interaction at all. Course moderators cannot make any intervention because they cannot enter into those groups.

Regarding the group composition's impact, through observing course participants' dropout and learning performance, the algorithmically composed groups reveal their capability of reducing the drop-out rate in comparison with a random grouping condition and a control condition (no grouping). However, the learning performance among those three conditions is not significantly different.

- **RQ3:** When creating small learning groups in the large scale learning settings (e.g. MOOCs), what new problems could arise and is there an approach to mitigate those problems?

The growing number of students in the recent large-scale learning environments challenges the efficiency of the group composition methods. The exact methods could take hours or even days to make a group formation for thousands of students.

Leveraging the group composition algorithms could make things more efficient (cf. answers to the RQ2). The group composition algorithms demand students' data. No algorithm can thus far runs in a cold-start mode. Different than classroom students or lab participants, the majority of MOOC students are, however, reluctant to expose their data. This leads to a situation in which the group composition algorithms merely work for those comparatively few data contributors. For the majority of students without data, a random method would not be a bad choice.

Another foreseeable problem is that the created small groups will face difficulties in remaining complete owing to the high drop-out rate. Many learning groups might have drop-out students more or less. This, in the end, poses a serious problem of human resource loss in those groups.

To mitigate the mentioned two problems, this thesis proposes a dynamic group re-composition approach (cf. Section 4.1). The proposed approach, in fact, composes learning groups multiple times rather than one time. Thus, students in the problematic groups, especially incomplete in size, can find a chance to reassign themselves into a new group. Since the approach additionally makes use of another source of data: group interaction data, the scarcity of data contribution could be lessened to some extent. One could argue that there could still be very few students participating in group discussions, but one more data source is, in principal, better than the single source.

- **RQ4:** If there is an approach, how can one put it into practice and what benefits could this approach bring?

The proposed dynamic group re-composition approach is data-driven. It requires group interaction data and group outcome data. Yet, most of the current MOOC platforms do not facilitate a channel to collect such group-related data. To deal with the provision of such data, a group tool was developed. This tool facilities small group discussions and group writing (see details in Section 4.2 and 4.3). Integrating this tool with the MOOC courses, students are assigned into small learning groups and re-composed into new groups if they would like.

With regard to the benefits of the proposed approach, a computer simulation experiment was conducted. In that experiment, drop-out and group cohesion were observed. The results show that the group re-composition approach can reduce the drop-out rate and bring more cohesive learning groups than a random condition (see details in Chapter 5).

6.2 Limitations

Regarding limitations of this research, first of all, the proposed group re-composition approach has not been evaluated in a real MOOC course. All the findings regarding the approach are extracted from a computer simulation system and thus are very sensitive to the simulation settings. Although the simulation system parameterizes the key factors in MOOCs (e.g. the drop-out rate), properly setting those parameters requires much more empirical evidence. To address this, it demands more discussion on the effect of tuning those parameters over a realistically possible range. Owing to the lack of a test in a real case, it could cast doubt on the approach's operability. For example, the group tool organizes students to do peer grading because we need their group performance data. The reality could turn out to be very little participation in such an event. This could badly affect the validation of the proposed approach because the data source is almost cut off then. The alternative is that we can impute much data by guessing (e.g. sampling based on the known average learning performance). But this can only solve the technical problem, that is, the tool would not get stuck and stop at some point. Still, how to boost students' participation is an open question.

Second, the dynamics of students' team roles might need a closer look. Currently, the group tool updates the latest role of each student. For instance, if a student plays a *leader* role in Task 1 but a *broker* in Task 2, then *broker* is taken as his role for the upcoming task. In fact, this student could be able to play both roles in groups. Different group circumstances may trigger his potentials to play a more proper role than the other. In this sense, simply updating the latest role for each student could not be a good choice. If allowing multiple roles for each student, the computational complexity will grow accordingly. One more role of each single student means one more role combination possibility for the whole group.

Third, the group composition rules learned are currently updated task by task. That means that it always leverages the latest group composition rules to form groups for the upcoming task. The composition rules in the previous tasks are thrown away. This action certainly strips away much knowledge of group composition. A better choice would be accumulating such group composition rules task by task but there would then be many repetitive rules or even contradictory rules. This will need an additional '*judge*' to justify the rules and trim the repetitive or inferior group composition rules.

Last but not least, with regard to the use of the twoleaves group tool, little has been discussed from the teachers' perspective. Teachers actually play a vitally important role in implementing the method. They need to assign group tasks to MOOC students. If they assign fairly easy tasks, many students can do it alone and do not need a group at all. If the assigned tasks are too difficult, many students would not be able to solve them. There could be such a concern from MOOC teachers. To solve this problem, the tool or the MOOC providers may need to offer a pre-course program, in which students can be taught how to collaborate with others. In the meantime, teachers can seize this chance to better feel their students' group skills and thus can design more achievable group tasks for their students (Brindley, Blaschke, & Walti, 2009).

6.3 Outlook

This work hopes to inspire some new thinking to explore the current body of group composition research. The field of group composition has experienced a history of random grouping and criteria-guided grouping. The future could go to a scope of data intelligence where we do not need the pre-defined grouping criteria anymore. The grouping criteria can directly be induced from students' behavioral data. Nevertheless, the doubt cast on the justification of the grouping criteria could still exist like in the case of the pre-defined grouping criteria. But by leveraging data intelligence, the induced group criteria mirror the specific learning contexts and thus avoid the over-generalization problem that the pre-defined grouping criteria could have.

The recent studies point out many difficulties when implementing small group learning in an open learning environment. Many students tell their interest but few of them actually participate in group events. On the one hand, we need to accept this fact. Online students have a wide range of motivations to take the course. They should take the most efficient strategy they want. On the other hand, to confront this problem, a pre-course training might be necessary. Many students may need to learn team skills (e.g. how to arrange team events and how to make group solutions). Such a pre-course training could help them with team skills and critical thinking. Additionally, this procedure can help with pre-selecting team work lovers and targeting more stable data contributors for the future use.

A look at the group dissatisfaction issue might be interesting too. As far as teamwork goes, it is believed that such problems as dissatisfaction and conflicts would arise more or less. In traditional classrooms, teacher may occasionally hear some of these too.

They sometimes have to adjust their students' positions in groups if some complaints are too loud to be ignored anyhow. In MOOCs, this could probably occur as well. Thereby, offering a channel for students to express their dissatisfaction with their team members might be reasonable. Facing a large-scale learning environment, teachers could not be able to make interventions to resolve such conflicts one by one. A walk-around could be that the group composition algorithm avoids assignments of those dissatisfied peers into the same group. This, in the end, requires some improvements from the algorithm's side.

Lastly, what must be done in the very near future is to test the twoleaves tool and validate the proposed method in real MOOC courses. Of greatest interest is to evaluate its impact on learning and teaching in the long run. Additionally, reducing the gap between the simulation system and the MOOC courses is of importance as well. Collecting much more evidence from the empirical studies and feeding them into the simulation system can achieve this goal. When the simulation system very closely approaches to the real situation, we can perhaps then conduct some experiments simply on the simulator rather than the real course so that the experimental lifecycle can be greatly cut down and unexpected influencing factors can also be under control.

7 REFERENCES

Anderson, T., & Dron, J. (2010). Three generations of distance education pedagogy. *The International Review of Research in Open and Distributed Learning, 12*(3).

Aviv, R., Erlich, Z., Ravid, G., & Geva, A. (2003). Network analysis of knowledge construction in asynchronous learning networks. *Journal of Asynchronous learning networks, 7*(3), 1-23.

Axelrod, R. (1997). Advancing the Art of Simulation in the Social Sciences. In R. Conte, R. Hegselmann, & P. Terna (Eds.), *Simulating Social Phenomena* (pp. 21-40). Berlin, Heidelberg: Springer Berlin Heidelberg.

Azzopardi, L., & Maxwell, D. (2017). *How To Tango With Django 1.7*: retrieved from: http://www.tangowithdjango.com/book17/chapters/setup.html#creating-your-django-project.

Bacon, D. R., Stewart, K. A., & Anderson, E. S. (2001). Methods of Assigning Players to Teams: A Review and Novel Approach. *Simulation & Gaming, 32*(1), 6-17. doi:10.1177/104687810103200102

Balakrishnan, G. (2013). *Predicting student retention in massive open online courses using hidden markov models.* Master's thesis, EECS Department, University of California, Berkeley.

Balmaceda, J., Schiaffino, S., & Pace, A. D. (2014). Using constraint satisfaction to aid group formation in CSCL. *Inteligencia artificial: Revista Iberoamericana de Inteligencia Artificial, 17*(53), 35-45.

Barrick, M. R., Stewart, G. L., Neubert, M. J., & Mount, M. K. (1998). Relating member ability and personality to work-team processes and team effectiveness. *Journal of Applied Psychology, 83*(3), 377-391. doi:10.1037/0021-9010.83.3.377

Beane, W. E., & Lemke, E. A. (1971). Group variables influencing the transfer of conceptual behavior. *Journal of Educational Psychology, 62*(3), 215-218. doi:10.1037/h0031145

Belbin, R. M. (1981). *Management Teams: Why They Succeed Or Fail*: Butterworth-Heinemann.

Belbin, R. M. (2010). *Team Roles at Work*: Butterworth-Heinemann.

Bergey, P., & King, M. (2014). Team Machine: A Decision Support System for Team Formation. *Decision Sciences Journal of Innovative Education, 12*(2), 109-130. doi:10.1111/dsji.12027

Bradley, J. H., & Hebert, F. J. (1997). The effect of personality type on team performance. *Journal of Management Development, 16*(5), 337-353. doi:doi:10.1108/02621719710174525

Bratitsis, T., & Dimitracopoulou, A. (2006). Monitoring and Analyzing Group Interactions in Asynchronous Discussions with the DIAS System. In Y. Dimitriadis, I. Zigurs, & E. Gómez-Sánchez (Eds.), *Groupware: Design, Implementation, and Use* (Vol. 4154, pp. 54-61): Springer Berlin Heidelberg.

Brindley, J., Blaschke, L. M., & Walti, C. (2009). Creating Effective Collaborative Learning Groups in an Online Environment. *The International Review of Research in Open and Distributed Learning, 10*(3).

Capuano, N., Mangione, G. R., Mazzoni, E., Miranda, S., & Orciuoli, F. (2014). Wiring role taking in collaborative learning environments. SNA and semantic web can improve CSCL script. *International Journal of Emerging Technologies in Learning (iJET), 9*(7), 30-38.

Chang, C.-K., Chen, G.-D., & Wang, C.-Y. (2011). Statistical model for predicting roles and effects in learning community. *Behaviour & Information Technology, 30*(1), 101-111. doi:10.1080/0144929X.2010.516017

Changsheng, Z., Jigui, S., Yan, W., & Qingyun, Y. (2007, 5-12 Nov. 2007). *An Improved Discrete Particle Swarm Optimization Algorithm for TSP.* Paper presented at the Web Intelligence and Intelligent Agent Technology Workshops, 2007 IEEE/WIC/ACM International Conferences on.

Chapman, K. J., Meuter, M., Toy, D., & Wright, L. (2006). Can't We Pick our Own Groups? The Influence of Group Selection Method on Group Dynamics and Outcomes. *Journal of Management Education, 30*(4), 557-569. doi:10.1177/1052562905284872

Colorni, A., Dorigo, M., & Maniezzo, V. (1991). *Distributed optimization by ant colonies.* Paper presented at the Proceedings of the first European conference on artificial life.

Colorni, A., Dorigo, M., & Maniezzo, V. (1992). *An Investigation of some Properties of an ``Ant Algorithm".* Paper presented at the PPSN.

Croes, G. A. (1958). A Method for Solving Traveling-Salesman Problems. *Operations Research, 6*(6), 791-812. doi:doi:10.1287/opre.6.6.791

Cruz, W. M., & Isotani, S. (2014). Group Formation Algorithms in Collaborative Learning Contexts: A Systematic Mapping of the Literature. In N. Baloian, F. Burstein, H. Ogata, F. Santoro, & G. Zurita (Eds.), *Collaboration and Technology: 20th International Conference, CRIWG 2014, Santiago, Chile, September 7-10, 2014. Proceedings* (pp. 199-214). Cham: Springer International Publishing.

Davis, J. P., Eisenhardt, K. M., & Bingham, C. B. (2007). Developing Theory through Simulation Methods. *The Academy of Management Review, 32*(2), 480-499.

Decker, R. (1995). Management team formation for large scale simulations. *Developments in Business Simulation and Experiential Learning, 22*.

Dooley, K. (2002). Simulation research methods. *Companion to organizations*, 829-848.

Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *Evolutionary Computation, IEEE Transactions on, 1*(1), 53-66. doi:10.1109/4235.585892

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 26*(1), 29-41. doi:10.1109/3477.484436

Dorigo, M., & Stützle, T. (2009). *Ant Colony Optimization: Overview and Recent Advances*. Retrieved from

Downes, S. (2008). Places to go: Connectivism & connective knowledge. *Innovate: Journal of Online Education, 5*(1), 6.

Duran, O., Rodriguez, N., & Consalter, L. A. (2008, 23-24 Jan. 2008). *A PSO-Based Clustering Algorithm for Manufacturing Cell Design.* Paper presented at the Knowledge Discovery and Data Mining, 2008. WKDD 2008. First International Workshop on.

Eberhart, R., & Kennedy, J. (1995, 4-6 Oct 1995). *A new optimizer using particle swarm theory.* Paper presented at the Micro Machine and Human Science, 1995. MHS '95., Proceedings of the Sixth International Symposium on.

Eiben, A. E., Hinterding, R., & Michalewicz, Z. (1999). Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation, 3*(2), 124-141. doi:10.1109/4235.771166

Eiben, A. E., & Smith, J. E. (2010). *Introduction to evolutionary computing* (Vol. 2): Springer Berlin.

Erlin, Rahmiati, & Rio, U. (2014). *Content Analysis to Detect The Role Behaviors of Student in Online Discussion*. Paper presented at the Proceeding of the Electrical Engineering Computer Science and Informatics.

Evans, C. R., & Dion, K. L. (1991). Group Cohesion and Performance: A Meta-Analysis. *Small Group Research, 22*(2), 175-186. doi:10.1177/1046496491222002

Felder, R. M., & Silverman, L. K. (1988). Learning and teaching styles in engineering education. *Engineering education, 78*(7), 674-681.

Feng, T., Shibin, W., Cheng, Z., & Qinghua, Z. (2008, 16-18 April 2008). *Research on e-learner personality grouping based on fuzzy clustering analysis.* Paper presented at the Computer Supported Cooperative Work in Design, 2008. CSCWD 2008. 12th International Conference on.

Gedeon, T. D., & Turner, S. (1993, 25-29 Oct. 1993). *Explaining student grades predicted by a neural network.* Paper presented at the Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan).

Gogoulou, A., Gouli, E., Boas, G., Liakou, E., & Grigoriadou, M. (2007). Forming homogeneous, heterogeneous and mixed groups of learners. *Proc. ICUM*, 33-40.

Graesser, A. C., Wiemer-Hastings, K., Wiemer-Hastings, P., & Kreuz, R. (1999). AutoTutor: A simulation of a human tutor. *Cognitive Systems Research, 1*(1), 35-51. doi:http://dx.doi.org/10.1016/S1389-0417(99)00005-4

Graf, S., & Bekele, R. (2006). *Forming heterogeneous groups for intelligent collaborative learning systems with ant colony optimization.* Paper presented at the Proceedings of the 8th international conference on Intelligent Tutoring Systems, Jhongli, Taiwan.

Grefenstette, J. J. (1986). Optimization of Control Parameters for Genetic Algorithms. *IEEE Transactions on Systems, Man, and Cybernetics, 16*(1), 122-128. doi:10.1109/TSMC.1986.289288

Gullahorn, J. T., & Gullahorn, J. E. (1964). *Computer simulation of human interaction in small groups*. Paper presented at the Proceedings of the April 21-23, 1964, spring joint computer conference, Washington, D.C.

Hansen, M. T. (1999). The Search-Transfer Problem: The Role of Weak Ties in Sharing Knowledge across Organization Subunits. *Administrative Science Quarterly, 44*(1), 82-111. doi:10.2307/2667032

Hoos, H. H., & Stützle, T. (2005). 8 - TRAVELLING SALESMAN PROBLEMS. In H. H. H. Stützle (Ed.), *Stochastic Local Search* (pp. 357-416). San Francisco: Morgan Kaufmann.

Hsu, C.-C., Chen, H.-C., Huang, K.-H., Huang, K.-K., & Huang, Y.-M. (2014). The development of an adaptive group composition system on Facebook for collaborative learning using an artificial bee colony algorithm. *Appl. Math, 8*(1L), 157-164.

Hsu, C.-C., Chen, H.-C., Huang, K.-K., & Huang, Y.-M. (2012). A personalized auxiliary material recommendation system based on learning style on Facebook applying an artificial bee colony algorithm. *Computers & Mathematics with Applications, 64*(5), 1506-1513. doi:10.1016/j.camwa.2012.03.098

Hsu, C. C., Chen, H. C., Chang, C. H., & Huang, Y. M. (2012). A Learning Companion Recommendation System on Facebook Using an Artificial Bee Colony Algorithm. *J. Internet Technol.*(13), 817–826.

Hurley, T., & Weibelzahl, S. (2007). Using MotSaRT to Support On-Line Teachers in Student Motivation. In E. Duval, R. Klamma, & M. Wolpers (Eds.), *Creating New Learning Experiences on a Global Scale: Second European Conference on Technology Enhanced Learning, EC-TEL 2007, Crete, Greece, September 17-20, 2007. Proceedings* (pp. 101-111). Berlin, Heidelberg: Springer Berlin Heidelberg.

Hwang, G. J., Yin, P. Y., Hwang, C. W., & Tsai, C. C. (2008). An enhanced genetic approach to composing cooperative learning groups for multiple grouping criteria. *Educational Technology & Society, 11*(1), 148-167.

Johnson, D. (1990). Local optimization and the Traveling Salesman Problem. In M. Paterson (Ed.), *Automata, Languages and Programming* (Vol. 443, pp. 446-461): Springer Berlin Heidelberg.

Johnson, D. S., & McGeoch, L. A. (1997). The traveling salesman problem: A case study in local optimization. *Local search in combinatorial optimization, 1*, 215-310.

Johnson, G. (2006). Synchronous and Asynchronous Text-Based CMC in Educational Contexts: A Review of Recent Research. *TechTrends, 50*(4), 46-53. doi:10.1007/s11528-006-0046-9

Jordan, K. (2014). MOOC Completion Rates: The Data. Retrieved 27/08/2014 http://www.katyjordan.com/MOOCproject.html

Jozan, M. M. B., Taghiyareh, F., & Faili, H. (2012). An Inversion-based Genetic Algorithm for Grouping of students. *The 7th International Conference on Virtual Learning ICVL 2012*.

Kinghorn, B. H. (2013). Computer Simulation: What's the Story? *American Journal of Management, 13*(2), 22.

Kirschner, P. A. (2015). Facebook as learning platform: Argumentation superhighway or dead-end street? *Computers in Human Behavior, 53*, 621-625. doi:http://dx.doi.org/10.1016/j.chb.2015.03.011

Kizilcek, R. F. (2013). *Collaborative Learning in Geographically Distributed and In-person Groups*. Paper presented at the 1st Workshop on Massive Open Online Courses at the 16th Annual Conference on Artificial Intelligence in Education, Memphis, TN., .

Kliegl, J. A., & Weaver, K. D. (2013). Teaching Teamwork Through Coteaching in the Business Classroom. *Business Communication Quarterly*. doi:10.1177/1080569913507596

Kloft, M., Stiehler, F., Zheng, Z., & Pinkward, N. (2014). *Predicting MOOC Dropout over Weeks Using Machine Learning Methods.* Paper presented at the Proceedings of the 2014 Empirical Methods in Natural Language Processing Workshop on Modeling Large Scale Social Interaction in Massively Open Online Courses.

Konert, J., Burlak, D., & Steinmetz, R. (2014). The Group Formation Problem: An Algorithmic Approach to Learning Group Formation *Open Learning and Teaching in Educational Communities* (pp. 221-234): Springer.

Koppenhaver, G. D., & Shrader, C. B. (2003). Structuring the Classroom for Performance: Cooperative Learning with Instructor-Assigned Teams. *Decision Sciences Journal of Innovative Education, 1*(1), 1-21.

Lechner, C., Frankenberger, K., & Floyd, S. W. (2010). Task Contingencies in the Curvilinear Relationships Between Intergroup Networks and Initiative Performance. *Academy of Management Journal, 53*(4), 865-889. doi:10.5465/amj.2010.52814620

Leskovec, J., Backstrom, L., Kumar, R., & Tomkins, A. (2008). *Microscopic evolution of social networks*. Paper presented at the Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, Las Vegas, Nevada, USA.

Lin, Y.-T., Huang, Y.-M., & Cheng, S.-C. (2010). An automatic group composition system for composing collaborative learning groups using enhanced particle swarm optimization. *Computers & Education, 55*(4), 1483-1493. doi:10.1016/j.compedu.2010.06.014

Lingard, R. W. (2010). Teaching and assessing teamwork skills in engineering and computer science. *Journal of Systemics, Cybernetics and Informatics, 18*(1), 34-37.

Lu, F., Li, X., Liu, Q., Yang, Z., Tan, G., & He, T. (2007). Research on Personalized E-Learning System Using Fuzzy Set Based Clustering Algorithm. In Y. Shi, G. D. van Albada, J. Dongarra, & P. M. A. Sloot (Eds.), *Computational Science – ICCS 2007: 7th International Conference, Beijing, China, May 27 - 30, 2007, Proceedings, Part III* (pp. 587-590). Berlin, Heidelberg: Springer Berlin Heidelberg.

Manca, S., & Ranieri, M. (2013). Is it a tool suitable for learning? A critical review of the literature on Facebook as a technology-enhanced learning environment. *Journal of Computer Assisted Learning, 29*(6), 487-504. doi:10.1111/jcal.12007

Marcos-García, J.-A., Martínez-Monés, A., & Dimitriadis, Y. (2015). DESPRO: A method based on roles to provide collaboration analysis support adapted to the participants in CSCL situations. *Computers & Education, 82*(0), 335-353. doi:http://dx.doi.org/10.1016/j.compedu.2014.10.027

Marcos-Garcıa, J. A., Martınez-Monés, A., Dimitriadis, Y., & Rodrıguez-Triana, M. J. (2008). Role-AdaptIA: A role-based adaptive tool for interaction analysis.

Marra, R. M., Moore, J. L., & Klimczak, A. K. (2004). Content analysis of online discussion forums: A comparative analysis of protocols. *Educational Technology Research and Development, 52*(2), 23-40.

Matsuda, N., Cohen, W. W., Sewall, J., Lacerda, G., & Koedinger, K. R. (2007a). *Evaluating a Simulated Student Using Real Students Data for Training and Testing^{\ thanks {The research presented in this paper is supported by National Science Foundation Award No. REC-0537198.}}*. Paper presented at the International Conference on User Modeling.

Matsuda, N., Cohen, W. W., Sewall, J., Lacerda, G., & Koedinger, K. R. (2007b). Predicting students' performance with simstudent: Learning cognitive skills from observation. *Frontiers in Artificial Intelligence and Applications, 158*, 467.

MOOCs@Edinburgh-Group. (2013). *MOOCs @ Edinburgh 2013 – Report #1*. Retrieved from http://hdl.handle.net/1842/6683

Moreno, J., Ovalle, D. A., & Vicari, R. M. (2012). A genetic algorithm approach for group formation in collaborative learning considering multiple student characteristics. *Computers & Education, 58*(1), 560-569. doi:10.1016/j.compedu.2011.09.011

Morgeson, F. P., Reider, M. H., & Campion, M. A. (2005). SELECTING INDIVIDUALS IN TEAM SETTINGS: THE IMPORTANCE OF SOCIAL SKILLS, PERSONALITY CHARACTERISTICS, AND TEAMWORK KNOWLEDGE. *Personnel Psychology, 58*(3), 583-611. doi:10.1111/j.1744-6570.2005.655.x

Newman, D. R., Webb, B., & Cochrane, C. (1995). A content analysis method to measure critical thinking in face-to-face and computer supported group learning. *Interpersonal Computing and Technology, 3*(2), 56-77.

Nguyen Thi Ngoc, H., & Haddawy, P. (2007, 10-13 Oct. 2007). *A decision support system for evaluating international student applications.* Paper presented at the 2007 37th Annual Frontiers In Education Conference - Global Engineering: Knowledge Without Borders, Opportunities Without Passports.

Nygren, E. (2010). *Modeling the social dynamics of online discussion sites.* Paper presented at the Proceedings of the International Workshop on Modeling Social Media, Toronto, Ontario, Canada.

Nygren, E. (2011). Grooming Analysis Modeling the Social Interactions of Online Discussion Groups. In M. Atzmueller, A. Hotho, M. Strohmaier, & A. Chin (Eds.), *Analysis of Social Media and Ubiquitous Data: International Workshops MSM 2010, Toronto, Canada, June 13, 2010, and MUSE 2010, Barcelona, Spain, September 20, 2010, Revised Selected Papers* (pp. 37-56). Berlin, Heidelberg: Springer Berlin Heidelberg.

Nygren, E. (2012). Simulation of User Participation and Interaction in Online Discussion Groups. In M. Atzmueller, A. Chin, D. Helic, & A. Hotho (Eds.), *Modeling and Mining Ubiquitous Social Media: International Workshops MSM 2011, Boston, MA, USA, October 9, 2011, and MUSE 2011, Athens, Greece, September 5, 2011, Revised Selected Papers* (pp. 138-157). Berlin, Heidelberg: Springer Berlin Heidelberg.

Onah, D., Sinclair, J., & Boyatt, R. (2014). DROPOUT RATES OF MASSIVE OPEN ONLINE COURSES: BEHAVIOURAL PATTERNS. *EDULEARN14 Proceedings*, 5825-5834.

Ounnas, A., Davis, H. C., & Millard, D. E. (2009). A Framework for Semantic Group Formation in Education. *Educational Technology & Society, 12*(4), 43-55.

Oztok, M., Zingaro, D., Brett, C., & Hewitt, J. (2013). Exploring asynchronous and synchronous tool use in online courses. *Computers & Education, 60*(1), 87-94. doi:http://dx.doi.org/10.1016/j.compedu.2012.08.007

P2PU. (2014). How we Used the Echo Nest API for Engagement & Learning. Retrieved from http://info.p2pu.org/2014/01/13/how-we-used-the-echonest-api-for-engagement-learning/

Padilha, E., & Carletta, J. (2002). *A simulation of small group discussion.* Paper presented at the Proceedings of EDILOG.

Padilha, E. G. (2006). Modelling turn-taking in a simulation of small group discussion.

Paredes, P., Ortigosa, A., & Rodriguez, P. (2010). A Method for Supporting Heterogeneous-Group Formation through Heuristics and Visualization. *J. UCS, 16*(19), 2882-2901.

Piech, C., Huang, J., Chen, Z., Do, C., Ng, A., & Koller, D. (2013). *Tuned Models of Peer Assessment in MOOCs*. Paper presented at the 6th International Conference on Educational Data Mining (EDM 2013).

Qiang, L., & Xiaoyan, X. (2007, 24-27 Aug. 2007). *Towards Termination Criteria of Ant Colony Optimization.* Paper presented at the Natural Computation, 2007. ICNC 2007. Third International Conference on.

Rabbany, R., Takaffoli, M., & Zaïane, O. R. (2011). *Analyzing participation of students in online courses using social network analysis techniques.* Paper presented at the Proceedings of educational data mining.

Ranjan, J., & Khalil, S. (2008). Conceptual framework of data mining process in management education in India: an institutional perspective. *Information Technology Journal, 7*(1), 16-23.

Reeves, C. R. (1995). A genetic algorithm for flowshop sequencing. *Computers & Operations Research, 22*(1), 5-13. doi:http://dx.doi.org/10.1016/0305-0548(93)E0014-K

Rosé, C. P., & Siemens, G. (2014). *Shared task on prediction of dropout over time in massively open online courses*. Paper presented at the 2014 Conference on Empirical Methods In Natural Language Processing Workshop on Modeling Large Scale Social Interaction In Massively Open Online Courses, Doha, Qata, .

Rutherfoord, R. H. (2001). *Using personality inventories to help form teams for software engineering class projects.* Paper presented at the ACM SIGCSE Bulletin.

Safe, M., Carballido, J., Ponzoni, I., & Brignole, N. (2004). On Stopping Criteria for Genetic Algorithms. In A. L. C. Bazzan & S. Labidi (Eds.), *Advances in Artificial Intelligence – SBIA 2004: 17th Brazilian Symposium on Artificial Intelligence, Sao Luis, Maranhao, Brazil, September 29-Ocotber 1, 2004. Proceedings* (pp. 405-413). Berlin, Heidelberg: Springer Berlin Heidelberg.

Senior, B. (1997). Team roles and team performance: Is there 'really' a link? *Journal of Occupational and Organizational Psychology, 70*(3), 241-258. doi:10.1111/j.2044-8325.1997.tb00646.x

Sharan, S. E. (1994). *Handbook of cooperative learning methods*: Greenwood Press/Greenwood Publishing Group.

Shimazoe, J., & Aldrich, H. (2010). Group Work Can Be Gratifying: Understanding & Overcoming Resistance to Cooperative Learning. *College Teaching, 58*(2), 52-57. doi:10.1080/87567550903418594

Siemens, G. (2004). Connectivism: A Learning Theory for the Digital Age.

Siemens, G. (2006a). Connectivism: Learning theory or pastime of the self-amused. *Manitoba, Canada: Learning Technologies Centre*.

Siemens, G. (2006b). *Knowing Knowledge*.

Siemens, G. (2014). Connectivism: A learning theory for the digital age.

Slavin, R. E. (1983). When does cooperative learning increase student achievement? *Psychological bulletin, 94*(3), 429.

Slavin, R. E. (1987). Ability Grouping and Student Achievement in Elementary Schools: A Best-Evidence Synthesis. *Review of Educational Research, 57*(3), 293-336. doi:10.3102/00346543057003293

Srba, I., & Bielikova, M. (2014). Dynamic Group Formation as an Approach to Collaborative Learning Support. *Learning Technologies, IEEE Transactions on, PP*(99). doi:10.1109/TLT.2014.2373374

Stahl, G. (2015). *Essays in Group-Cognitive Science volume#10*.

Stasser, G. (1988). Computer simulation as a research tool: The DISCUSS model of group decision making. *Journal of Experimental Social Psychology, 24*(5), 393-422. doi:http://dx.doi.org/10.1016/0022-1031(88)90028-5

Stevens Jr, K. T. (1998). *The effects of roles and personality characteristics on software development team effectiveness.* (Ph.D thesis), Faculty of Virginia Polytechnic Institute and State University.

Stuetzer, C. M., Koehler, T., Carley, K. M., & Thiem, G. (2013). “Brokering” Behavior in Collaborative Learning Systems. *Procedia - Social and Behavioral Sciences, 100*, 94-107. doi:http://dx.doi.org/10.1016/j.sbspro.2013.10.702

Su, Z., Song, W., Lin, M., & Li, J. (2008, 12-14 Dec. 2008). *Web Text Clustering for Personalized E-learning Based on Maximal Frequent Itemsets.* Paper presented at the 2008 International Conference on Computer Science and Software Engineering.

Sundararajan, B. (2010). Emergence of the most knowledgeable other (MKO): Social network analysis of chat and bulletin board conversations in a CSCL system. *Electronic Journal of e-Learning, 8*(2), 191-208.

Tsai, W., & Ghoshal, S. (1998). Social Capital and Value Creation: The Role of Intrafirm Networks. *Academy of Management Journal, 41*(4), 464-476. doi:10.2307/257085

Tuckman, B. W. (1965). Developmental sequence in small groups. *Psychological bulletin, 63*(6), 384-399. doi:10.1037/h0022100

Tuckman, B. W., & Jensen, M. A. C. (1977). Stages of Small-Group Development Revisited. *Group & Organization Management, 2*(4), 419-427. doi:10.1177/105960117700200404

Vogelsang, T., & Ruppertz, L. (2015). *On the validity of peer grading and a cloud teaching assistant system*. Paper presented at the Proceedings of the Fifth International Conference on Learning Analytics And Knowledge, Poughkeepsie, New York.

Wasserman, S. (1994). *Social network analysis: Methods and applications* (Vol. 8): Cambridge university press.

Webb, N. M. (1982). Student Interaction and Learning in Small Groups. *Review of Educational Research, 52*(3), 421-445. doi:10.3102/00346543052003421

Wen, M., Maki, K., Wang, X., Dow, S. P., Herbsleb, J., & Rose, C. (2016). *Transactivity as a Predictor of Future Collaborative Knowledge Integration in Team-Based Learning in Online Courses.* Paper presented at the Educational Data Mining, NC, US.

Wen, M., Yang, D., & Rosé, C. P. (2015). Virtual Teams in Massive Open Online Courses. In C. Conati, N. Heffernan, A. Mitrovic, & F. M. Verdejo (Eds.), *Artificial Intelligence in Education: 17th International Conference, AIED 2015, Madrid, Spain, June 22-26, 2015. Proceedings* (pp. 820-824). Cham: Springer International Publishing.

White, K. B. (1984). MIS project teams: An investigation of cognitive style implications. *MIS Quarterly, 8*, 95-101.

Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and Computing, 4*(2), 65-85. doi:10.1007/bf00175354

Whittingham, K. L. (2006). Impact of personality on academic performance of MBA students: Qualitative versus quantitative courses. *Decision Sciences Journal of Innovative Education, 4*(2), 175-190.

Wichmann, A., Hecking, T., Elson, M., Christmann, N., Herrmann, T., & Hoppe, H. U. (2016). *Group Formation for Small-Group Learning: Are Heterogeneous Groups More Productive?* Paper presented at the OpenSym '16, Berlin, Germany.

Wise, A. F., Zhao, Y., & Hausknecht, S. N. (2013). *Learning analytics for online discussions: a pedagogical model for intervention with embedded and extracted analytics*. Paper presented at the Proceedings of the Third International Conference on Learning Analytics and Knowledge, Leuven, Belgium.

Wise, S. (2014). Can a team have too much cohesion? The dark side to network density. *European Management Journal, 32*(5), 703-711. doi:http://dx.doi.org/10.1016/j.emj.2013.12.005

Woeginger, G. J. (2003). Exact algorithms for NP-hard problems: A survey. *Lecture notes in computer science, 2570*(2003), 185-207.

Yang, D., Wen, M., & Rose, C. (2014). *Peer Influence on Attrition in Massively Open Online Courses*. Paper presented at the 7th International Conference on Educational Data Mining, London, UK.

Yannibelli, V., & Amandi, A. (2012a). A deterministic crowding evolutionary algorithm to form learning teams in a collaborative learning context. *Expert Systems with Applications, 39*(10), 8584-8592. doi:http://dx.doi.org/10.1016/j.eswa.2012.01.195

Yannibelli, V., & Amandi, A. (2012b). A Memetic Algorithm for Collaborative Learning Team Formation in the Context of Software Engineering Courses. In F. Cipolla-Ficarra, K. Veltman, D. Verber, M. Cipolla-Ficarra, & F. Kammüller (Eds.), *Advances in New Technologies, Interactive Interfaces and Communicability* (Vol. 7547, pp. 92-103): Springer Berlin Heidelberg.

Zheng, Z., & Pinkwart, N. (2014, 7-10 July 2014). *A Discrete Particle Swarm Optimization Approach to Compose Heterogeneous Learning Groups.* Paper presented at the Advanced Learning Technologies (ICALT), 2014 IEEE 14th International Conference on.

Zheng, Z., Stapel, M., & Pinkwart, N. (2016). *Perfect Scores Indicate Good Students!? The Case of One Hundred Percenters in a Math Learning System*. Paper presented at the 9th International Conference on Educational Data Mining, Raleigh, NC, US.

Zheng, Z., Vogelsang, T., & Pinkwart, N. (2015). *The Impact of Small Learning Group Composition on Student Engagement and Success in a MOOC*. Paper presented at the 8th International Conference on Educational Data Mining (EDM 2015), Madrid, Spain, .

Zielinski, K., & Laur, R. (2007). Stopping Criteria for a Constrained Single-Objective Particle Swarm Optimization Algorithm. *Informatica (03505596), 31*(1).

Zielinski, K., & Laur, R. (2008). Stopping Criteria for Differential Evolution in Constrained Single-Objective Optimization. In U. Chakraborty (Ed.), *Advances in Differential Evolution* (Vol. 143, pp. 111-138): Springer Berlin Heidelberg.

Ziv, A., Wolpe, P. R., Small, S. D., & Glick, S. (2006). Simulation-Based Medical Education: An Ethical Imperative. *Simulation in Healthcare, 1*(4), 252-256. doi:10.1097/01.sih.0000242724.08501.63

8 APPENDICES

APPENDIX Ⅰ: MOTIVATION SURVEY QUESTION (THE SELECTED ONE OF IVERSITY'S MOTIVATION SURVEY QUESTIONS)

The following is the question used to collect student's presumed video consumption data. It was selected from the iversity's online motivation survey (Version 2.1).

Question: I intend to watch _____ of the provided lecture videos.

Answers (single-selection):

- ☐ Some
- ☐ About half
- ☐ Most
- ☐ All

APPENDIX Ⅱ: DEMOGRAPHIC SURVEY QUESTION (THE SELECTED ONE OF IVERSITY'S DEMOGRAPHIC SURVEY QUESTIONS)

The following is the question used to collect student's geographic data. It was selected from the iversity's online demographic survey (Version 2.1).

Question: In what country do you currently live?

Answers (single-selection):

- EH - Western Sahara
- KZ – Kazakhstan
- PH – Philippines
- ET – Ethiopia
- …(more countries to select)

APPENDIX III: GRID SEARCH EXPERIMENT FOR THE GENETIC ALGORITHM

The following are facts of the grid search experiment conducted to determine the parameters' values. Each combination of parameters values and the grouping quality it can accordingly yield are presented in the following table.

Parameter grids:

- Population size ($\boldsymbol{nPopu}$): 30 | 50
- Crossover rate($\boldsymbol{P_c}$): 0.5 | 0.7 | 0.9
- Mutation rate($\boldsymbol{P_m}$): 0.01 | 0.05 | 0.1

Stopping criteria:

- the minimal number of iterations is 50
- the maximal number of iteratins is 100
- if the latest 2/3 iterations do not improve the resutls, it then stops.

Results:

Population size ($\boldsymbol{nPopu}$)	Crossover rate($\boldsymbol{P_c}$)	Mutation rate($\boldsymbol{P_m}$)	Quality of the resulting group formation
30	0.5	0.01	0.5219
30	0.5	0.05	0.5228
30	0.5	0.1	0.5243
30	0.7	0.01	0.5235
30	0.7	0.05	0.5245
30	0.7	0.1	0.5241
30	0.9	0.01	0.5240
30	0.9	0,05	0.5239
30	0.9	0.1	0.5218
50	0.5	0.01	0.5231
50	0.5	0.05	0.5235
50	0.5	0.1	0.5245
50	0.7	0.01	0.5244

Population size ($nPopu$)	Crossover rate(P_c)	Mutation rate(P_m)	Quality of the resulting group formation
50	0.7	0.05	0.5237
50	0.7	0.1	0.5232
50	0.9	0.01	0.5258
50	0.9	0.05	0.5263
50	0.9	0.1	0.5260

APPENDIX Ⅳ: GRID SEARCH EXPERIMENT FOR THE DISCRETE-PSO ALGORITHM

The following are facts of the grid search experiment conducted to determine the parameters' settings. Each combination of parameters values and the grouping quality it can accordingly yield are presented in the following table.

Parameter grids:

- Population size ($\boldsymbol{nPopu}$): 30 | 50
- Inertial weight($\boldsymbol{\omega}$): 0.1 | 0.5 | 0.9
- Self-learning factor($\boldsymbol{c_1}$)): 0.1 | 0.5 | 0.9
- Social-learning factor($\boldsymbol{c_2}$): 0.1 | 0.5 | 0.9

Stopping criteria:

- the minimal number of iterations is 50
- the maximal number of iteratins is 100
- if the latest 2/3 iterations do not improve the resutls, it then stops.

Results:

Population size ($\boldsymbol{nPopu}$)	inertial weight($\boldsymbol{\omega}$)	self-learning factor($\boldsymbol{c_1}$)	social-learning factor($\boldsymbol{c_2}$)	Quality of the resulting group formation
30	0.1	0.1	0.1	0.5273
30	0.1	0.1	0.5	0.5299
30	0.1	0.1	0.9	0.5228
30	0.1	0.5	0.1	0.5225
30	0.1	0.5	0.5	0.5200
30	0.1	0.5	0.9	0.5232
30	0.1	0.9	0.1	0.5257
30	0.1	0.9	0.5	0.5246
30	0.1	0.9	0.9	0.5191
30	0.5	0.1	0.1	0.5190
30	0.5	0.1	0.5	0.5229
30	0.5	0.1	0.9	0.5201

Population size ($\boldsymbol{nPopu}$)	inertial weight($\boldsymbol{\omega}$)	self-learning factor($\boldsymbol{c_1}$)	social-learning factor($\boldsymbol{c_2}$)	Quality of the resulting group formation
30	0.5	0.5	0.1	0.5208
30	0.5	0.5	0.5	0.5201
30	0.5	0.5	0.9	0.5200
30	0.5	0.9	0.1	0.5199
30	0.5	0.9	0.5	0.5185
30	0.5	0.9	0.9	0.5189
30	0.9	0.1	0.1	0.5194
30	0.9	0.1	0.5	0.5192
30	0.9	0.1	0.9	0.5194
30	0.9	0.5	0.1	0.5186
30	0.9	0.5	0.5	0.5194
30	0.9	0.5	0.9	0.5206
30	0.9	0.9	0.1	0.5196
30	0.9	0.9	0.5	0.5188
30	0.9	0.9	0.9	0.5189
50	0.1	0.1	0.1	0.5291
50	0.1	0.1	0.5	0.5279
50	0.1	0.1	0.9	0.5236
50	0.1	0.5	0.1	0.5231
50	0.1	0.5	0.5	0.5271
50	0.1	0.5	0.9	0.5236
50	0.1	0.9	0.1	0.5267
50	0.1	0.9	0.5	0.5212
50	0.1	0.9	0.9	0.5242
50	0.5	0.1	0.1	0.5198
50	0.5	0.1	0.5	0.5189
50	0.5	0.1	0.9	0.5202
50	0.5	0.5	0.1	0.5234
50	0.5	0.5	0.5	0.5218
50	0.5	0.5	0.9	0.5194

Population size ($\boldsymbol{nPopu}$)	inertial weight($\boldsymbol{\omega}$)	self-learning factor($\boldsymbol{c_1}$)	social-learning factor($\boldsymbol{c_2}$)	Quality of the resulting group formation
50	0.5	0.9	0.1	0.5212
50	0.5	0.9	0.5	0.5197
50	0.5	0.9	0.9	0.5203
50	0.9	0.1	0.1	0.5193
50	0.9	0.1	0.5	0.5190
50	0.9	0.1	0.9	0.5196
50	0.9	0.5	0.1	0.5191
50	0.9	0.5	0.5	0.5192
50	0.9	0.5	0.9	0.5186
50	0.9	0.9	0.1	0.5189
50	0.9	0.9	0.5	0.5191
50	0.9	0.9	0.9	0.5196

APPENDIX V: GROUPING SURVEY

The following is the survey that was used to collect student's data for the first grouping experiment on iversity's MOOC platform (Course Name: The Fascination of Crystals and Symmetry; Year: 2014).

Question 1: Which of the follow services would you like to use for communication within your learning group? Multiple answers are possible. But please only select a service, if you already have an account and are familiar with this service.

Type: Multiple Choice

Answers:

- ☐ Skype
- ☐ Facebook
- ☐ Google+
- ☐ meetup.com
- ☐ Email

Question 2: What is your Skype name?

Type: Free text.

Answers: ____

Question 3: Where will you be during the course? Please select the closest country.

Type: Single Choice.

Answers:

- ☐ A list of Countries to select

Question 4: Where will you be during the course? Please select the closest city.

Type: Single Choice.

Answers:

- ☐ A list of main cities of the selected country.

Question 5: Which kind of grouping would you prefer?

Type: Single Choice.

Answers:

- I would have time for local meetings in the selected city.
- I would prefer virtual meeting with the selected technologies.

I prefer not be be part of a learning group.

Question 6: Which language could be the language of conversation within your learning group? If you would feel comfortable with many languages, you can select all of them.

Type: Multiple Choice.

Answers:

- English
- German
- French
- Spanish
- Italian

Question 7: Optional: Which other language could be the language of conversation within your learning group? If you would feel comfortable with many languages, you can select all of them.

Type: Multiple Choice.

Answers:

- A list of other Languages of the world.

Question 8: What is your gender?

Type: Single Choice.

Answers:

- Male
- Female
- Other
- Rather not say

Question 9: Which statement best reflects your goal for this course?

Type: Single Choice.

Answers:

- I want to get a general idea of the topic.
- I want to study the subject in depth.

Question 10: Choose the statement you would be most likely to say within a meeting of your learning group.

Type: Single Choice.

Answers:

- I'm not sure we're on the right track.
- Let's try to look at this another way.
- So here's what we've decided so far.
- Let's come back to this later if we have time.
- That's a great idea!
- We haven't heard from Katherine yet: I'd like to hear what you think.
- I don't think you're right, but we could also add...
- Are we all in agreement on this?
- We only have five minutes left, so we need to come to an agreement now!
- I'm not sure I agree, what are your reasons for saying that?
- I have an idea!
- We CAN do this!

Question 11: Please present yourself to the group within three(!) sentences. Only 150 words. -What is your background? -Why are you taking this course? -Please describe a very good learning group experience you already had in your life.

Type: Free Text.

Answers: ____

APPENDIX VI: HETEROGENEITY AND HOMOGENEITY DATA

The following data regards the heterogeneity and homogeneity that each algorithm can yield. Each algorithm ran ten times. The average over those ten times was selected to measure the heterogeneity and homogeneity.

Runtime	Discrete-PSO		GA		Adapted k-means Algo.		ACO	
	Hetero.	Homo.	Hetero.	Homo.	Hetero	Homo.	Hetero.	Homo.
#1	0.2055	0.1536	0.2076	0.1518	0.2268	0.1490	0.1774	0.0273
#2	0.2087	0.1563	0.2047	0.1536	0.2269	0.1459	0.1901	0.0390
#3	0.2094	0.1518	0.2033	0.1539	0.2272	0.1464	0.1761	0.0252
#4	0.2047	0.1540	0.2047	0.1559	0.2257	0.1407	0.1894	0.0399
#5	0.2049	0.1528	0.2033	0.1500	0.2261	0.1438	0.1901	0.0384
#6	0.2104	0.1513	0.2091	0.1585	0.2240	0.1483	0.1757	0.0226
#7	0.2043	0.1495	0.2043	0.1560	0.2279	0.1437	0.1773	0.0271
#8	0.2099	0.1566	0.2066	0.1531	0.2256	0.1444	0.1872	0.0351
#9	0.2055	0.1551	0.2053	0.1584	0.2253	0.1448	0.1776	0.0267
#10	0.2105	0.1538	0.2043	0.1583	0.2271	0.1421	0.1767	0.0242
Ave.	0.2073	0.1534	0.2053	0.1549	0.2262	0.1449	0.1817	0.0305
SD	0.0024	0.0021	0.0017	0.0028	0.0010	0.0024	0.0061	0.0064

Selbständigkeitserklärung

Ich erkläre, dass ich die Dissertation selbständig und nur unter Verwendung der von mir gemäß § 7 Abs. 3 der Promotionsordnung der Mathematisch-Naturwissenschaftlichen Fakultät, veröffentlicht im Amtlichen Mitteilungsblatt der Humboldt-Universität zu Berlin Nr. 126/2014 am 18.11.2014 angegebenen Hilfsmittel angefertigt habe.

Zhilin Zheng

Göttingen, den 20.09.2017